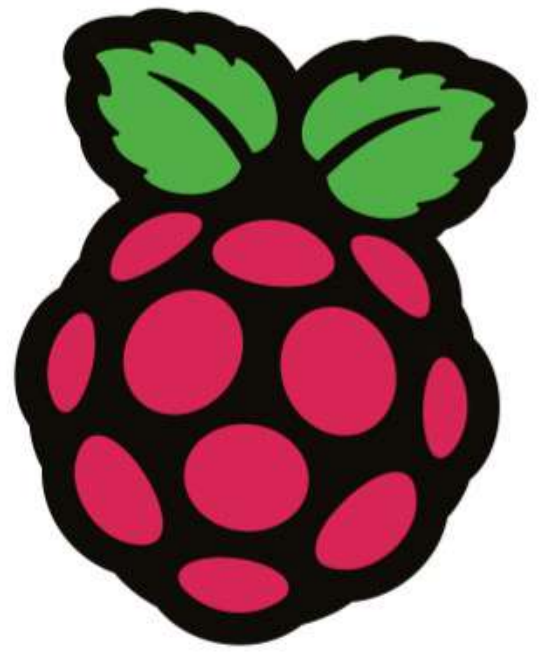




YOUR **OFFICIAL** RASPBERRY PI MAGAZINE

The MagPi



Issue 120

August 2022

magpi.cc

The official Raspberry Pi magazine

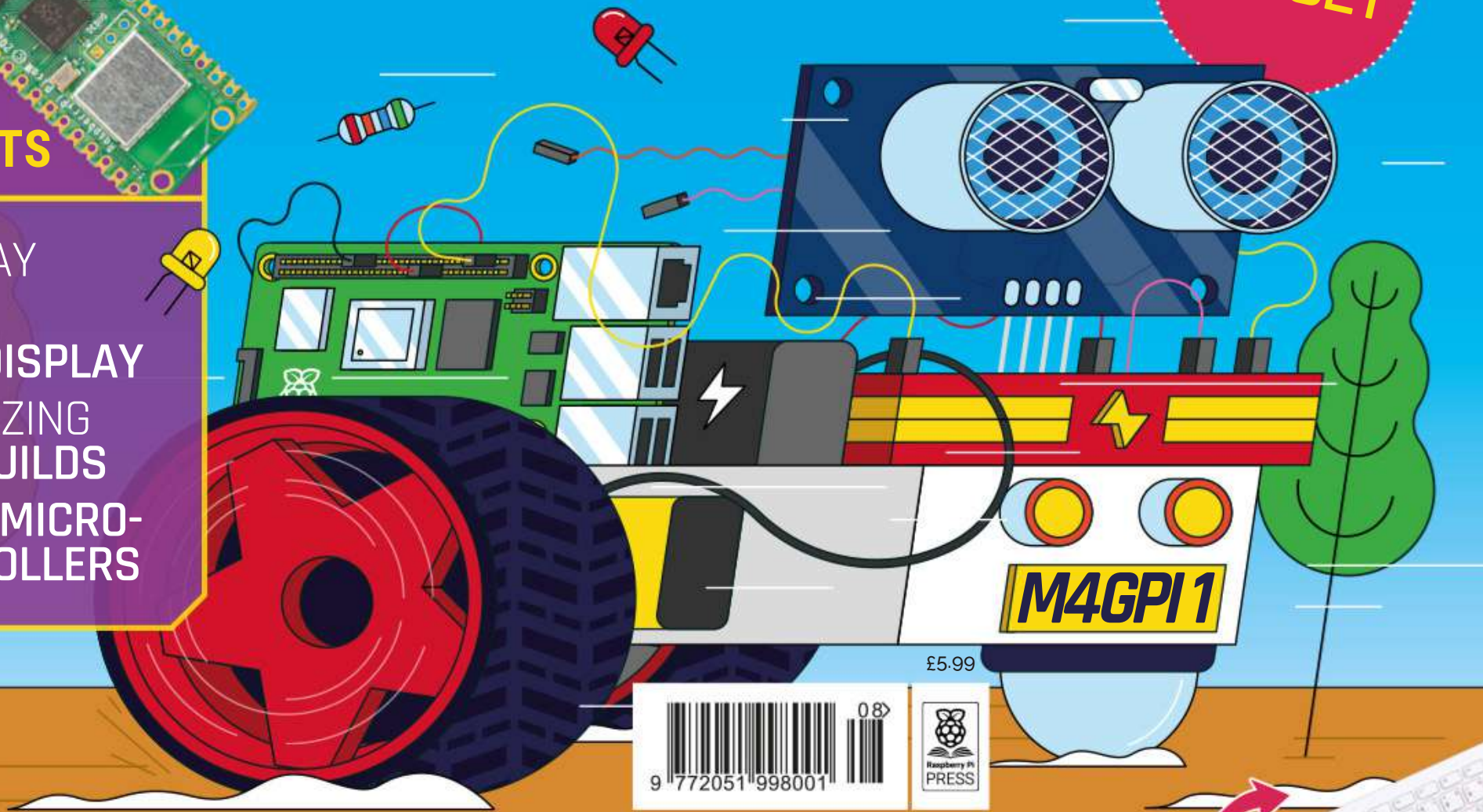
SUPER SIMPLE ROBOTICS

YES! YOU CAN BUILD THAT ROBOT

Build on a
BUDGET

PICO W PROJECTS

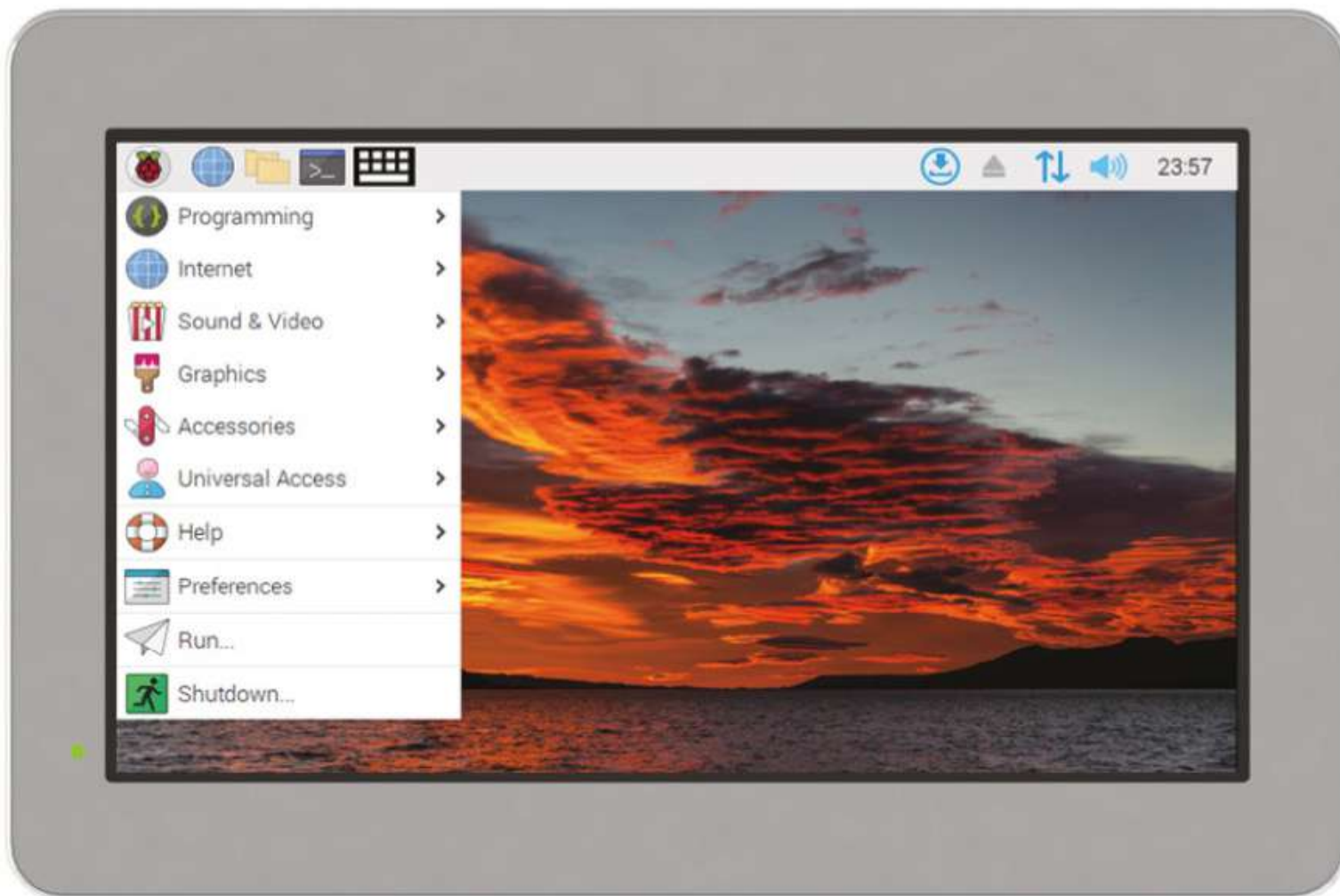
- > RAILWAY
CLOCK
- > E-INK DISPLAY
- > 10 AMAZING
PICO BUILDS
- > LEARN MICRO-
CONTROLLERS



BACK TO SCHOOL WITH **RASPBERRY PI!**



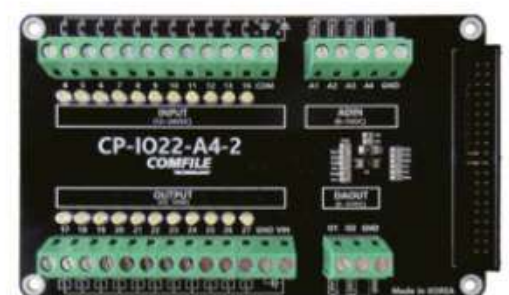
Industrial Raspberry Pi



ComfilePi

The ComfilePi is a touch panel PC designed with high-tolerant components and no moving parts for industrial applications. It features a water-resistant front panel, touchscreen, color LCD (available in various sizes), RS-232, RS-485, Ethernet, USB, I2C, SPI, digital IO, battery-backed RTC (real-time clock), and piezo buzzer.

Use the rear-panel 40-pin GPIO header to expand its features and capabilities with additional I/O boards. The ComfilePi is UL Listed and employs Raspberry Pi Compute Module.



WELCOME to The MagPi 120

I still remember my first robot: ‘Tyres’, named after the cyclist in *Spaced*. To this day, he’s a good bot! Building a robot from scratch is one of the most rewarding Raspberry Pi projects. So many people tell us that building a robot sounds incredibly complex. But it’s really not as hard as you first think.

Take one Raspberry Pi (or Pico), add a couple of motors, mix in a couple of wheels, and whizz it all up with a battery, and you will have a tin box whirring around the room in no time.

Of course, you can take it from there, and modern robot parts can become incredibly detailed. We’ve built all manner of walking, dancing, scuttling, and grabbing robots. All are controlled by code.

Raspberry Pi is about more than learning how to code and build robots. Our Back To School feature (page 60) is packed with advice on how to use Raspberry Pi to get the most out of your education. Don’t forget to build scuttling critters along the way.

Lucy Hattersley Editor



EDITOR **Lucy Hattersley**

Lucy routinely annoys her cat, Siouxsie to the garden and back, with robots of all manner (and some with no manners at all).

[@LucyHattersley](#)

GET A
**RASPBERRY PI
PICO W**
WITH A SUBSCRIPTION!
PAGE 30



it's summer...

get outside!

The pi-top **Robotics Superset** comes with everything you need for cool, **portable** projects, including:

- 50 + metal build pieces for unlimited creations
- Servo motors
- Wide-angle camera
- Sensors galore
- Your very own pi-top [4] with a 5-hour internal battery and OLED screen
- Access to the extensive Further project library
- So much more!

Your only limit is your imagination!



Learn more: www.pi-top.com

Buy now: shop.pi-top.com

pi-top

Contents

➤ Issue 120 ➤ August 2022

Cover Feature

32 Super Simple Robotics



Regulars

28 Case Study: PicoSystem

92 Your Letters

97 Next Month

98 The Final Word

Project Showcases

08 Pico Railway Clock

10 Pico PlayStation MemCard

14 SystemSix

16 VK-Pocket camera

20 Psion 5MX to Raspberry Pi Zero

22 Talkative Tube Dashboard

26 CogniFly



Pico PlayStation MemCard



CogniFly

The MagPi magazine is published monthly by Raspberry Pi Ltd. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The MagPi magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

Tutorials

- 44** Listen to birds with BirdNET-Pi
- 48** Build a 64-bit Minecraft server
- 52** Learn ARM Assembly - part 5

The Big Feature



Back to School 2022

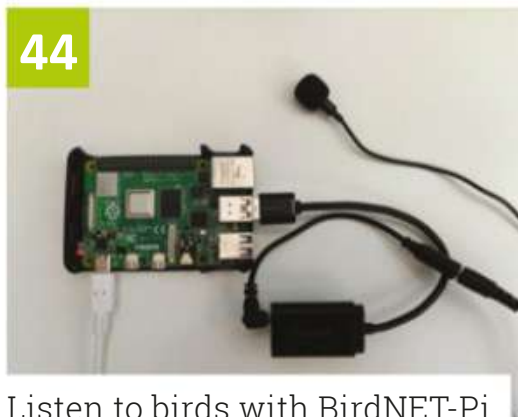
Reviews

- 68** CrowPi L
- 70** Motor SHIM for Pico
- 72** HiFiBerry DAC2 Pro & HD
- 74** Mini Pupper
- 76** Tindie Treats
- 82** Top 10 Pico Projects
- 84** Learn microcontrollers

Community

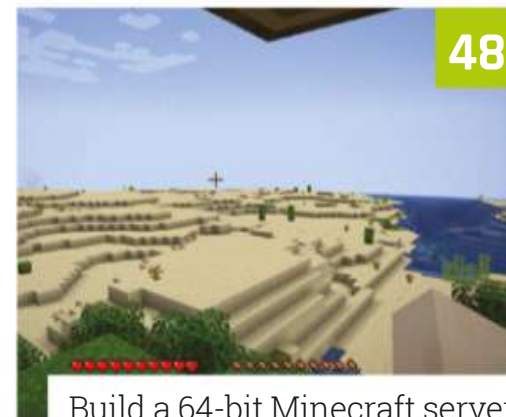
- 86** Matt Richardson interview
- 88** This Month in Raspberry Pi

44



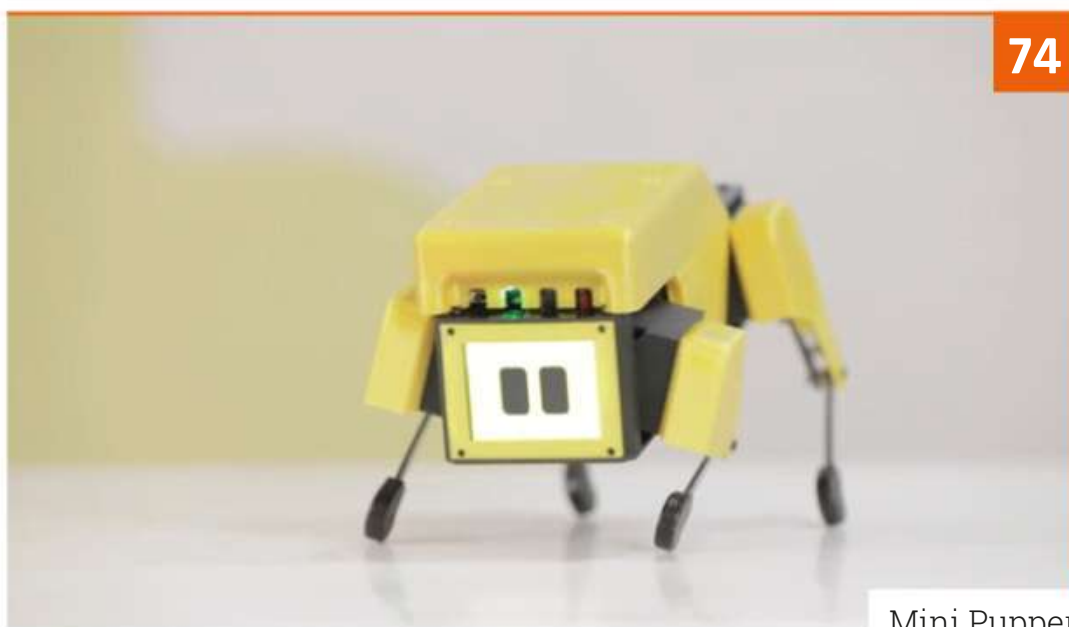
Listen to birds with BirdNET-Pi

48



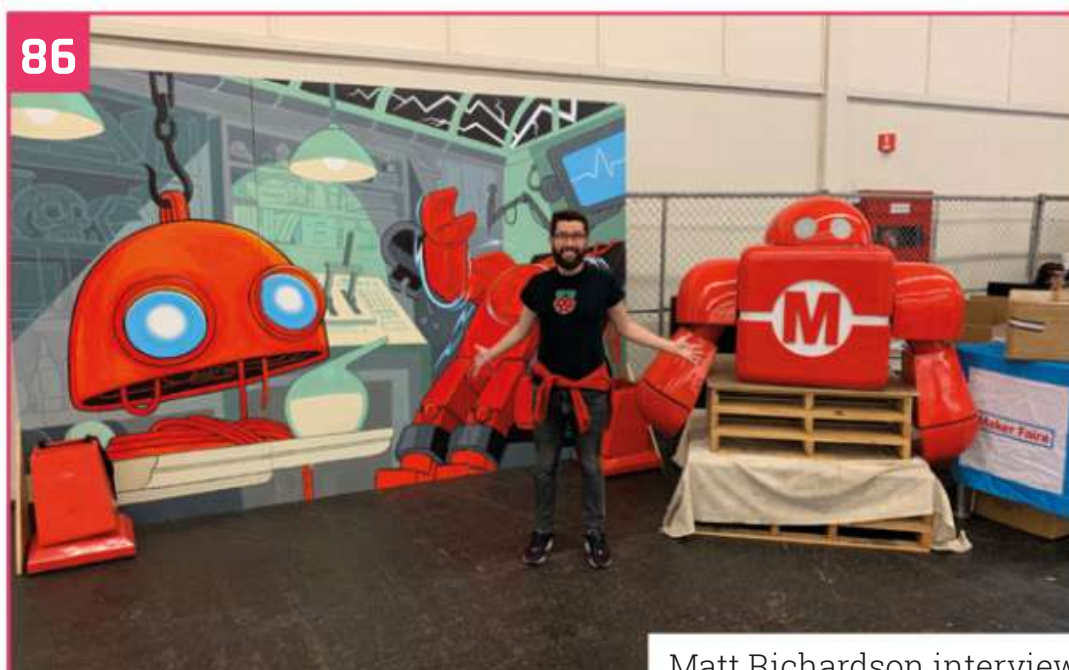
Build a 64-bit Minecraft server

74



Mini Pupper

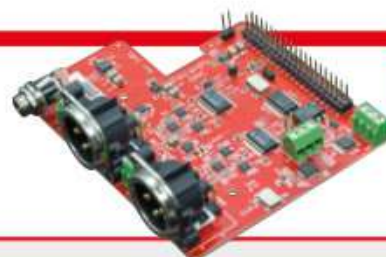
86



Matt Richardson interview

WIN A

PECANPI DAC
REV 3.0



95

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

INSPIRATION STARTS HERE



From millions of in-stock parts to cutting-edge technical resources—we've got everything you need to turn inspiration into innovation.

Get inspired at [digikey.co.uk](https://www.digikey.co.uk) or call 0800 587 0991.



Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

Pico Railway Clock

Giving new life to a clock from yesteryear. **Nicola King** celebrates the relaxing and reassuring sound of classically cool timepieces



MAKER

Martin Spendiff & Vanessa Bradley

Martin is a mathematical modeller who left the UK for Switzerland, and a fan of FOSS and tech that serves users, rather than the people who made it. Vanessa is new to coding and a constant source of weird and good ideas.

veeb.ch

Readers may remember a fantastic Teasmade project that we featured back in *The MagPi* issue #114 (magpi.cc/114), made by Swiss-based team Martin Spendiff and Vanessa Bradley. Well, while browsing their YouTube channel, we noticed that – in a new upcycling project – they have taken a vintage railway clock and transformed it with a Pico. Naturally, we wanted to know what made it tick.

Old timer

“We bought a railway station clock from a flea market and were a bit crestfallen when nothing happened when we plugged it in,” explains Martin. “The nice man who sold it seemed adamant that it worked, so after a bit of reading, I found out that it was waiting for a signal from a ‘Mutteruhr’.”

As the duo explain in their YouTube video (magpi.cc/railwaystationclock), often when you see such clock in a station setting, there is a delay between the second hand reaching 12 and the minute hand advancing; this is because the clock is waiting for an electrical pulse from the ‘Mutteruhr’, or mother/master clock. This pulse drives the minute hand forward and then the second hand is free to complete another cycle.

Martin and Vanessa had purchased what was essentially a secondary clock – ineffective without a mother clock. To get it working, they decided

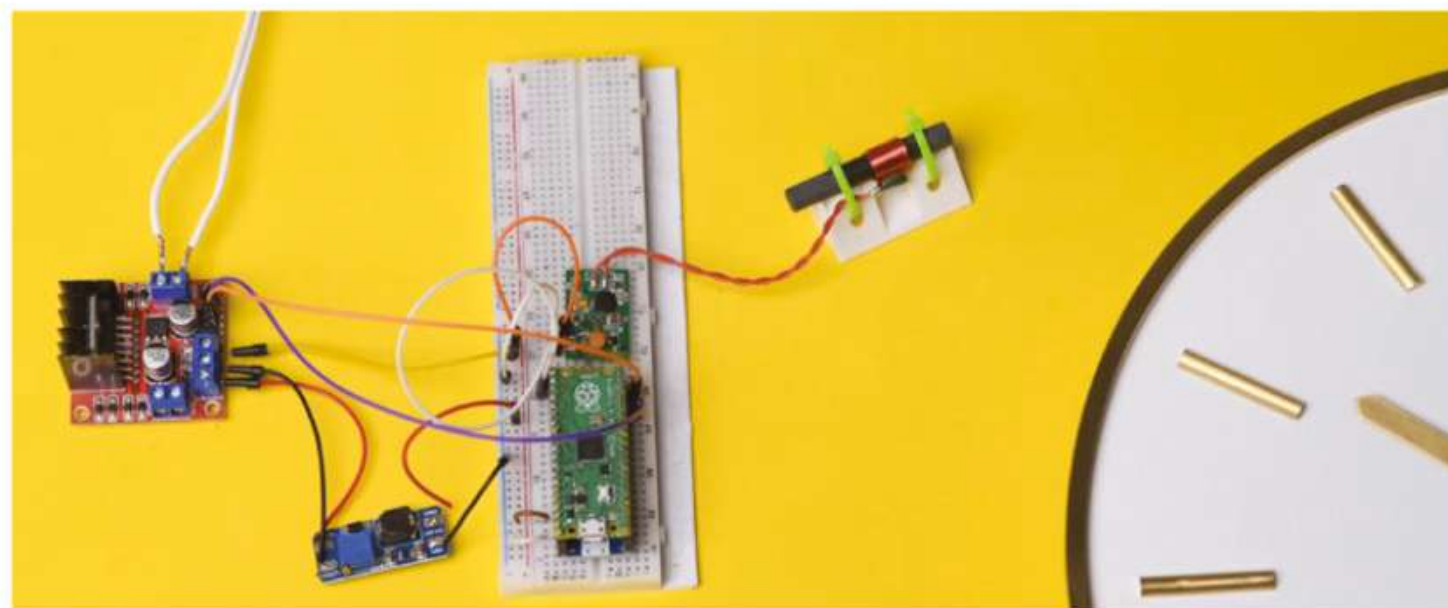
to build a mother clock themselves with a few additional components and some code running on a Raspberry Pi Pico microcontroller.

Inner workings

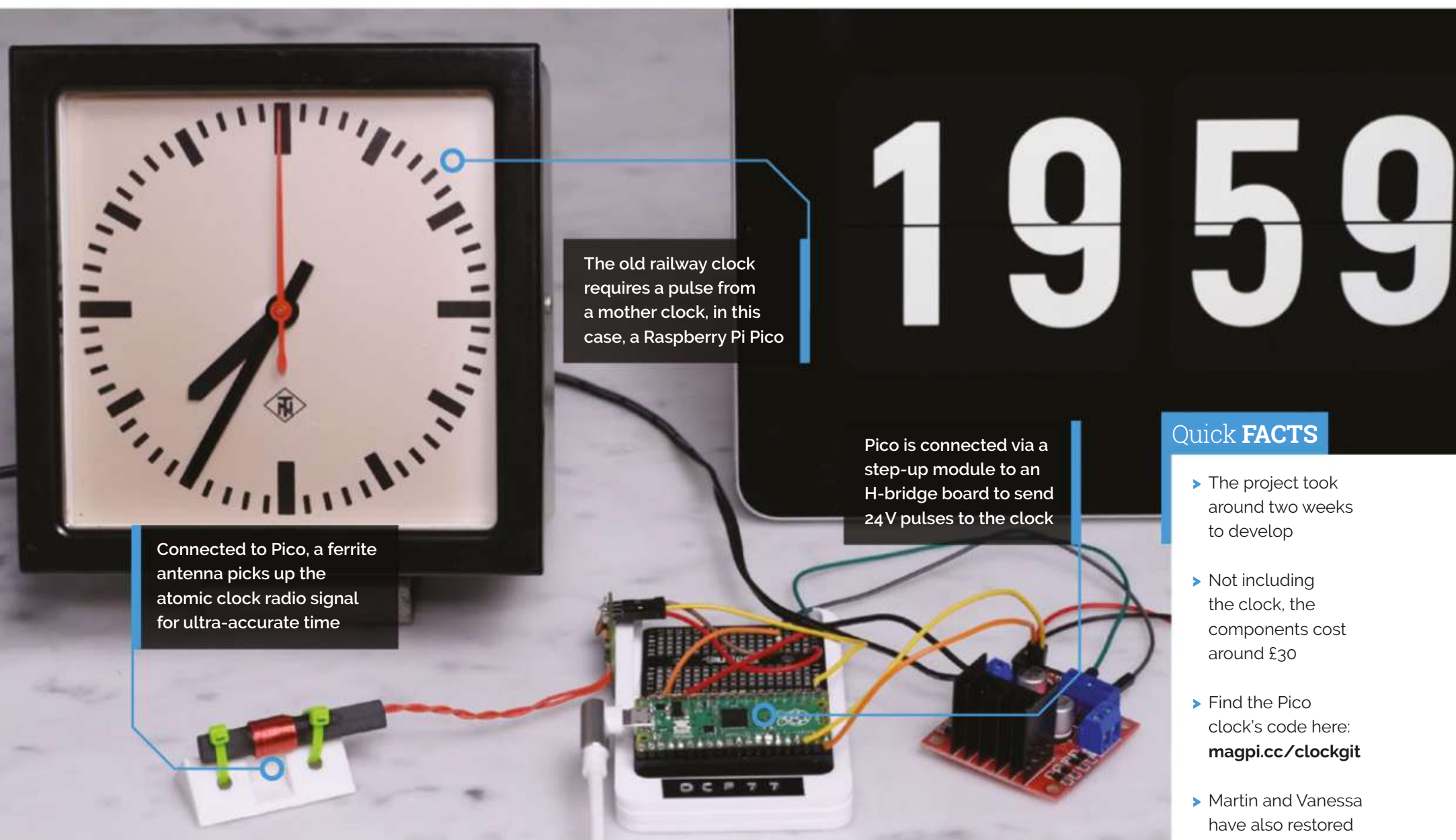
Attaching a ferrite antenna – to pick up the DCF77 atomic clock long-wave radio signals in Europe – to Pico was a first step, along with incorporating a real-time clock (RTC). “There is a text file that tells the code the time that is showing on the clock,” explains Martin. “You enter that manually. When Pico is plugged in, the code checks if the recorded time is the same as the RTC time – if not, it sends a pulse to the clock, updates the recorded time by a minute, and says ‘what about now?’ It just keeps doing that.”

The pair encountered few issues during the build and think it would be a relatively easy make to replicate. “The Python code needed to be tweaked a little, but it was relatively plain sailing,” says Martin, who reveals that they have now updated the code to also work in the US using the WWVB signal. A radio signal is not essential, however: “Setting the RTC manually would get it to work. The only difference is that the code would not update the RTC.”

The 3.3V output from Pico’s GPIO pins is converted to 24V by a step-up module, before being routed to an H-bridge to send the pulse to



▶ Wiring up the circuit. Components include an H-bridge, step-up module, and radio antenna



The old railway clock requires a pulse from a mother clock, in this case, a Raspberry Pi Pico

Pico is connected via a step-up module to an H-bridge board to send 24V pulses to the clock

Connected to Pico, a ferrite antenna picks up the atomic clock radio signal for ultra-accurate time

Quick **FACTS**

- ▶ The project took around two weeks to develop
- ▶ Not including the clock, the components cost around £30
- ▶ Find the Pico clock's code here: magpi.cc/clockgit
- ▶ Martin and Vanessa have also restored a German-made 1950s clock...
- ▶ You can see it in action on YouTube: magpi.cc/oldclock

the clock, although the voltage will depend on the timepiece used. "Some bigger clocks need a bigger electrical 'kick'," notes Martin.

“ I found out that it was waiting for a signal from a ‘Mutteruhr’ (mother clock) ”

The pair's interest in old clocks has led to quite the collection. Indeed, Martin admits they now have "too many", but he has a cunning plan to free up some wall space, as he says some will likely become birthday presents "for people that were foolish enough to look interested as we explained to them how they worked!"

As for upcoming ventures, they are certainly not short of spare horological parts. "We've got a box with 74 clock movements in it," reveals Martin. "We mentioned in the video that people often take the original movements out and replace them with quartz movements – we found one of those people, and convinced him to give us the leftovers." They are not entirely sure what they will do with all their clock components: "ideas welcome!"



▲ The rear of a similar vintage clock. When the polarity of its metal coil is flipped, a pair of magnets are repelled, turning the adjustment dials

Pico PlayStation MemCard

Failing hardware and pricey memory cartridges led one maker to use Raspberry Pi Pico to revive his old Sony PlayStation, as **Rosie Hattersley** discovers



MAKER

Daniele Giuliani

Daniele's passion for technology began while tinkering with Raspberry Pi at school in Italy, and he now holds an MSc in Computer Science.

magpi.cc/picomemcard



Warning!
Cut Safely

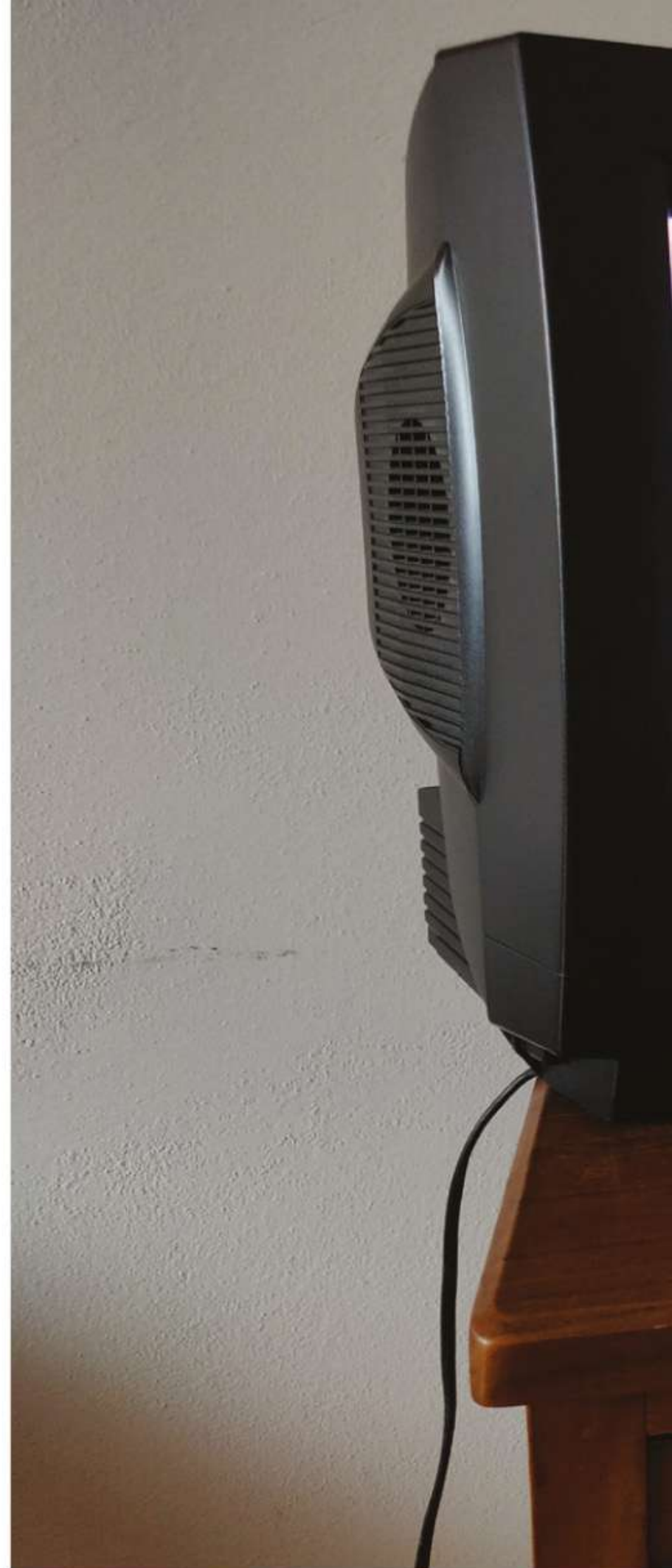
This project uses a sharp knife to cut open the original memory card. Watch over children when using sharp knives.

magpi.cc/knifesafety

Software engineer Daniele Giuliani missed playing some of the games on his original PlayStation, but quickly discovered that boxed and unused PS memory cards are hard to find and far more expensive than they used to retail for.

With no guarantee that used official cards – or counterfeit Chinese ones – will perform well, Daniele decided to put his coding skills to good use and create an alternative to Sony memory cards so he could continue playing his old PlayStation games. Having tinkered with Raspberry Pi devices since his high-school days, and subsequently studied computer science at university, Dan realised the platform would be ideal for his MemCard project.

Daniele chose a Raspberry Pi Pico because it's much faster than his beloved Raspberry Pi 1B+, which he still has. A fast GPIO connection was essential for this project. He was also delighted to discover Pico's 'novel' PIO (Programmable Input/Output) interface which he had never seen any board offer. "With PIO you can program, using specific assembly instructions, a set of 'state machines' to control the GPIO directly, leaving the main processor free to do other work," he enthuses. "PIO allows the creation of very powerful bus sniffers. I used it to program the low-level interaction, the basic signals that must be toggled on and off with very specific timings in order to convince the PlayStation into believing an original memory card is present."



Bonus features

Choosing Raspberry Pi for the build brought added benefits: the range of connectivity options improved on what could be done with original memory cards." In particular, since Raspberry Pi can be connected to a PC, it allows [the player] to easily import/export the save files from/to the PlayStation. This is useful to back up old saves and continue playing on emulators," explains Daniele. "PlayStation uses discs to load games. Old discs are full of scratches and can sometimes result in a game freezing in specific parts of the

Quick **FACTS**

- ▶ Daniele's first Raspberry Pi was a gift for a friend
- ▶ He bought his own when he realised how capable it was!
- ▶ He still owns – and uses – his trusty Raspberry Pi 1B+...
- ▶ It's in a mini greenhouse that sits under his desk...
- ▶ Measuring soil temperature for his super-hot chillies

Pico makes use of the PlayStation's 3.3V memory card connection. Since Pico draws very little power, it needs no additional power source

The PlayStation loads games from discs. This clever Pico-based project provides a cost-effective way to save and load game positions

Programmable Input/Output and fast GPIO data transfers made Pico the ideal choice of hardware and advantages over original console memory cards

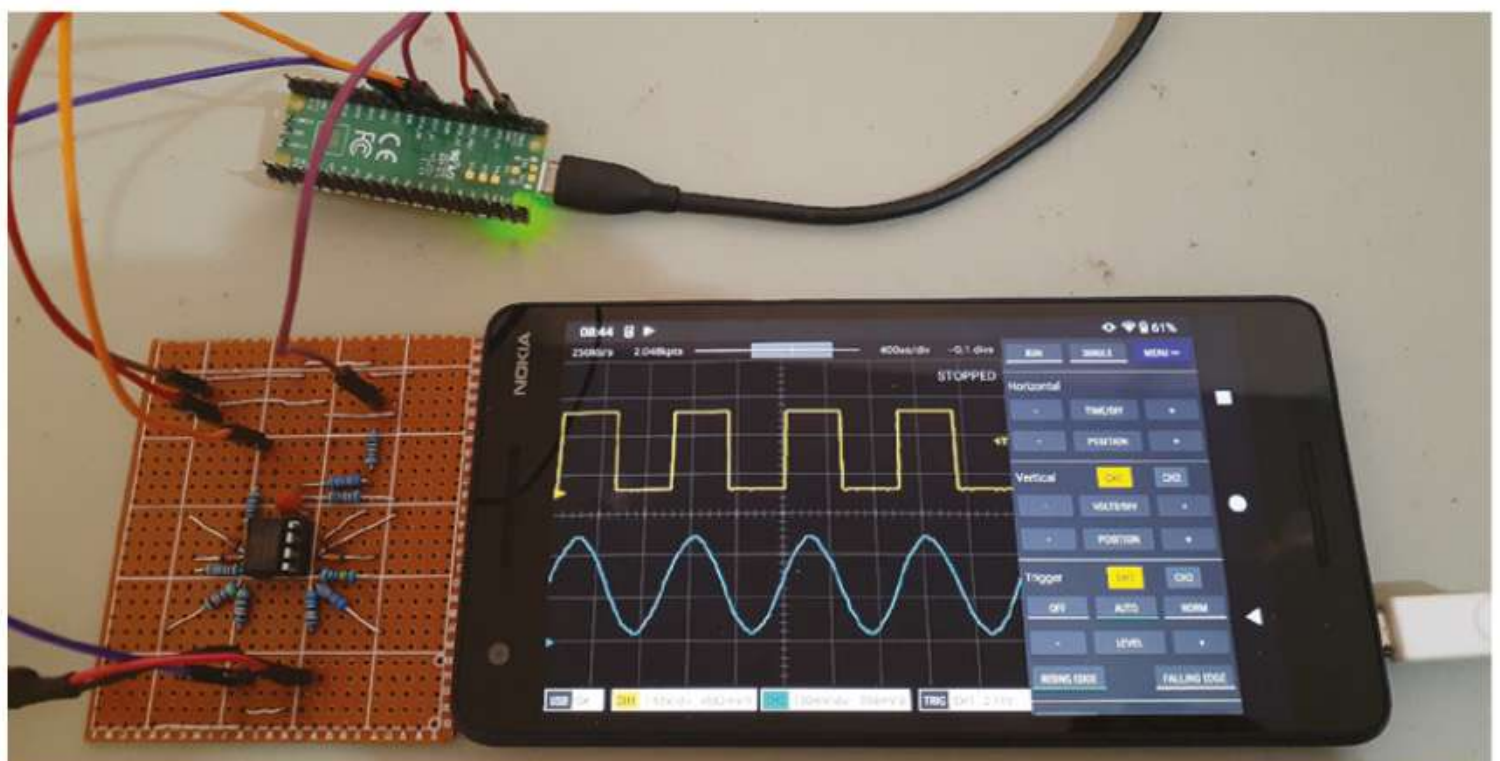
game, preventing the user from progressing.” With Daniele's Pico MemCard setup, a player can transfer the save file to an emulator, progress through the problematic area (using an intact backup image of the game), then transfer back the new save file and continue playing on the console.

The goal of this project was to provide a solid alternative at a very low price, says Daniele. “Raspberry Pi Pico understands the protocol used to communicate with memory cards and convinces the PlayStation into believing an original memory card is plugged in. Development boards, such as

“ Raspberry Pi Pico understands the protocol used to communicate with memory cards and convinces the PlayStation into believing an original memory card is plugged in ”



- ▲ The PlayStation recognises the Pico memory card and can save game data to it
- ▶ The Scopy oscilloscope app checks for signals between Pico and PlayStation






these ones, were particularly interesting because they were quite powerful yet very cheap, and allowed me to set up my personal server to test all sorts of applications.”

Smart tools

Although Daniele enjoys the freedom of working on his own projects, lack of equipment was a challenge. A smartphone app called Scopyy (magpi.cc/scopyy) proved invaluable: he was able to use a second Pico (plugged into a smartphone) as an oscilloscope to observe the electric signals being exchanged between the PlayStation and the memory card/Raspberry Pi Pico. Details that Daniele found online, of how a PlayStation works, helped him with some of the communication protocols. Prior to finding these details, he'd been trying to read the data between a PlayStation controller and console.

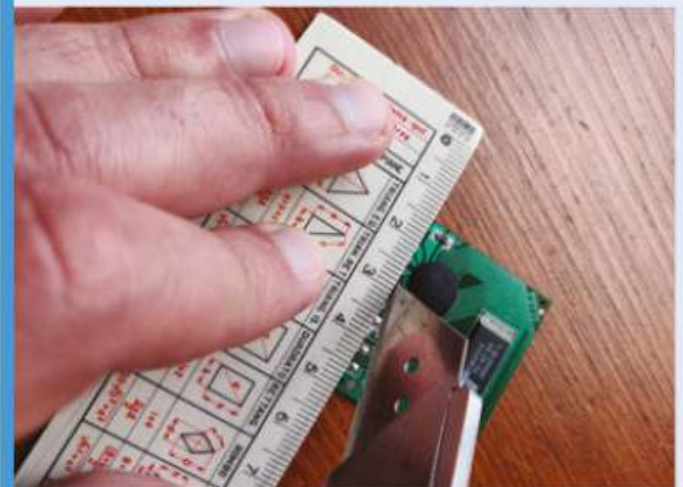
After a month or so of development for his MemCard for PlayStation, Daniele is already planning its next iteration: an enclosure for it, a microSD card version to expand the storage, and support for PS2. He's delighted with the amount of interest that others have shown in his project, and is proud of having found a practical way of reviving a classic, but otherwise unused, games console, potentially keeping them out of landfill. 

▲ PlayStation game saves from an emulator can be loaded from the Pico MemCard, enabling players to bypass bad sectors on game discs

Memory card maker



- 01** The substitute memory card provides a physical interface between the Pico and the PlayStation. Access the electronic board by removing the two screws on the bottom, and take off the plastic shell. Cut a hole in the case for the wires to connect your Pico.



- 02** Use a Stanley/utility knife to cut a groove right under the pins and disconnect the original circuitry. Use jumper wires to connect your Pico, scraping off their plastic covering, and directly soldering to the copper pins for the PlayStation.



- 03** Carefully solder your Pico to the PCB, download the latest version of PicoMemcard (magpi.cc/picomemcardreleases), plug your device into the computer via USB, and upload the save file.

SystemSix

John Calhoun has paid homage to his first Macintosh by creating an artistic-looking way to display useful information, as **David Crookes** explains



MAKER

John Calhoun

John started programming on a Mac, wrote shareware and commercial games, and worked for Apple for 26 years.

magpi.cc/systemsix

You may have heard of John Calhoun. Aside from working as a software engineer for Apple for 26 years until he retired last autumn, he's well-known for writing a string of games for the Apple Mac, among them the popular 1988 shareware title, Glider.

More recently, he's been causing a bit of a stir thanks to a different kind of fruit-related computer. By connecting a Raspberry Pi 3 to a 5.83-inch e-ink display, he's created a fresh spin on a classic Mac, called SystemSix – a modern homage to the System 6 operating system released 31 years ago.

Dubbed a “desk accessory”, it displays the local weather, a calendar, the current phase of the moon, and even the bin collection day, among other things. But what makes it extra-special is that it's housed on a stylish laser-cut piece of acrylic, bent to allow it to stand. It's an impressive piece of work.

Introducing SystemSix

John became fascinated with e-ink displays a while ago. One project which blew him away was a large display that resembled the front page of a newspaper, although it was very expensive. When he saw a cheaper e-ink project, however, he

was persuaded to give it a go. “I set out simply to recreate an e-ink calendar project I had seen on the web, and I also wanted to learn Python,” he says.

Having recreated the project, he began to modify the code, learning how and why it worked. “One of the early ideas was to add a lunar-phase to the calendar,” he says. “A quick search in GitHub showed many examples of an algorithm to determine the phase of the moon for a given date. I then needed two dozen or so images of the moon in various phases and did an image search.”

It then struck him that he'd have to convert the images to black and white. “Back in the day, when I was cutting my teeth in programming on a Macintosh Plus, [Apple coder and MacPaint author] Bill Atkinson had a fantastic algorithm that made very pleasant 1-bit image results,” he says. “I found dithering websites with an Atkinson option, and the results with the moon images were fantastic. Perfectly reminiscent of the retro Mac experience.”

It's that simple

The idea for a non-interactive desktop hub grew. “I had the moon displayed in black and white Atkinson pixels,” he says. “That was when the whole idea came to me: to go all in and make the entire calendar look like an early Mac. The current date could be displayed in a window title (perhaps the name of a document being edited). Maybe a Finder window would display a list of folders with the names of them corresponding to upcoming events...”

The task entailed many hours using Affinity Photo and Pixelmator to crop and clean up pieces of art, the idea being to replicate the look of Apple's System 6. Although it was always going to be a static, non-interactive project (you can't click on the screen or move the windows), he arranged them in layers to look like an early Mac desktop. “It wasn't too hard to drop them into the project and write code to render them,” he says. Yet he didn't stop there.

John also created different layouts. “I thought it would be fun to mix it up. Maybe a MacWrite layout



▶ John has also created a stand fitted with an LCD screen connected to a Raspberry Pi computer running the Mini vMac emulator



Raspberry Pi 3 is mounted at the back, running code written in Python

The stand is cut from a single sheet of 3 mm laser-cut acrylic, bent using a strip heater. It allows it to stand up while leaning back slightly

The Waveshare e-ink display's resolution of 648×480 pixels is larger than a Mac Plus's 512×342 pixels, hence a black border to lower it to 608×456 pixels

Quick **FACTS**

- ▶ SystemSix displays information daily on an e-ink screen
- ▶ It has been designed to resemble a Mac Plus
- ▶ Source artwork was taken from an emulator
- ▶ You could build one for about \$100
- ▶ All of the source code is on GitHub: magpi.cc/systemsixgh

one day, Finder with folders the next, and so on,” he explains, urging anyone replicating his project to try their hand at adding more. “Hypercard,

“ I had the moon displaying in black and white Atkinson pixels ”

a classic Mac app, could inspire a couple of new layouts,” he says. “I only wish animation on e-ink displays was better. An After Dark layout, if you remember that classic screensaver, would be very cool, although probably not on the cards.”



- ▶ As well as using a weather API, displaying calendar events, and displaying a full trash can on bin day, an Atkinson dithered moon shows its current phases

VK-Pocket camera

Long-time Raspberry Pi Zero fan James used one to create a Blade Runner-inspired camcorder, catching **Rosie Hattersley's** eye



James Brown

James lives in Wellington, New Zealand, has a background in games and graphics programming, and creates interactive exhibitions.

@ancientjames
@mastodon.
social

There's "something wonderfully unsettling" about being stared at by your own disembodied eye, comments master of understatement James Brown.

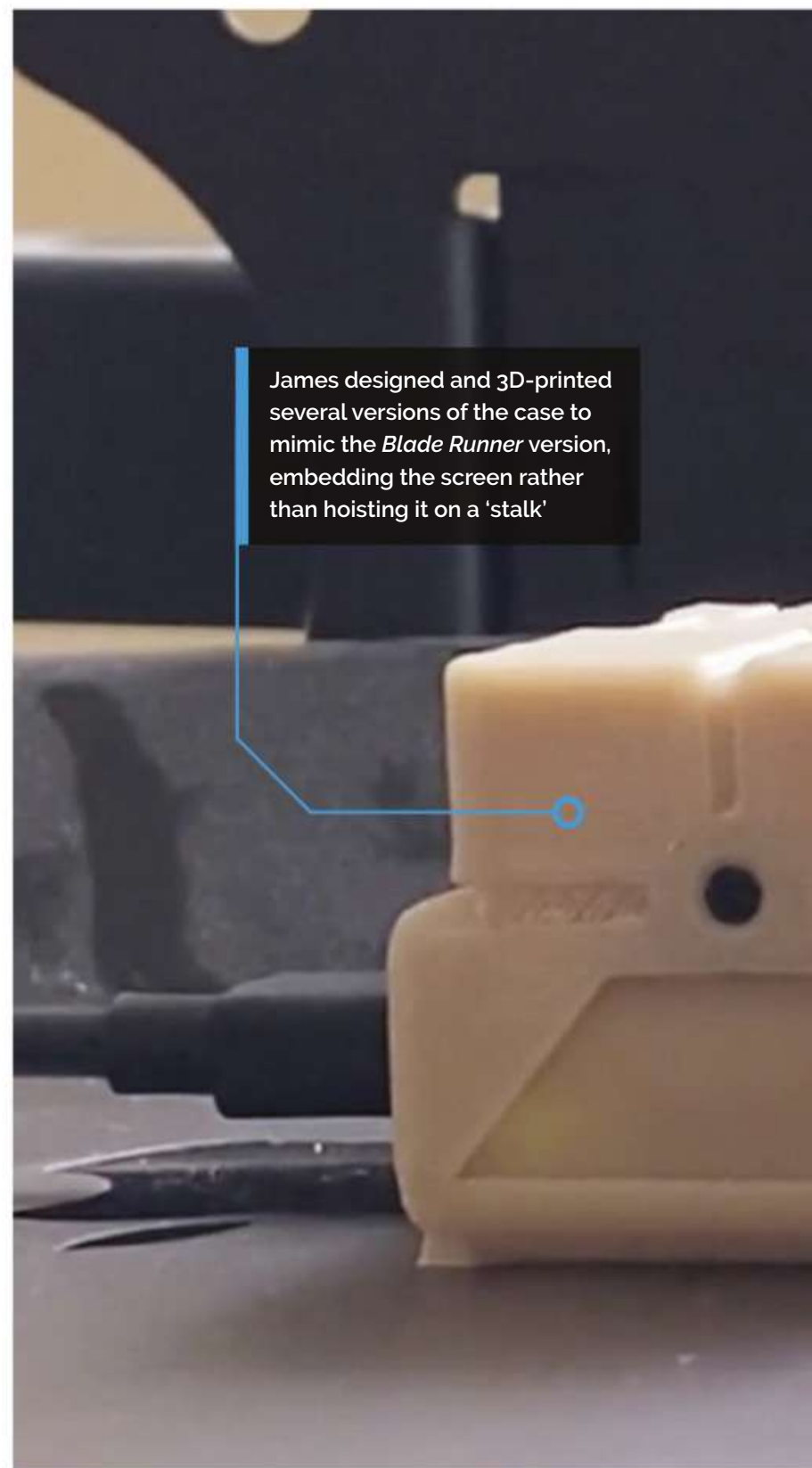
James came up with the idea for his VK-Pocket Camcorder while working on the face-tracking feature for an interactive exhibit, and realised one of the debug tools was showing parts of his face as it detected them. "I had a little viewfinder CRT salvaged from a junk shop camcorder, so I decided to wrap up that experience in a little self-contained gizmo, and style it after the Voight-Kampff machine from *Blade Runner*."

"In the movie, there's a camera on a stalk which is aimed at the subject's eyeball, and a monitor showing that eye isolated and magnified. My concept was to have a high-resolution, wide-angle camera, and use the face tracking code to crop and zoom in to any eye it detected. Anyone approaching the machine to look at it would be stared back at by their own eye."

Animated response

James knew immediately that he wanted to use Raspberry Pi Pico for his VK-Pocket camera project. Moreover, composite video out, which Pico supports, was essential for driving the CRT (cathode ray tube) display he culled from an old video camera. "Raspberry Pi Pico was my first choice for this build. I love these things", he exclaims! "They're a full Linux PC in a microcontroller form factor. I've put them in all sorts of builds, from animatronic heads to robotic insects." [Yes, we want to hear more about these projects, too - Ed].

James is a stickler for details so, as well as accommodating the mini screen, camera, and Pico, it was vital that the case for the homebrew VK machine looked like the original film prop.



James designed and 3D-printed several versions of the case to mimic the *Blade Runner* version, embedding the screen rather than hoisting it on a 'stalk'



Alert!
CRT

This project uses a CRT television. Be careful of high voltages when working with CRT screens.

magpi.cc/crt

Illustrating this is the "little servo" he added "to push some cosmetic bellows up and down," as a nod to those in the film. There are two versions of the VK machine in *Blade Runner*, he explains; "the device I ended up building is a bit of a mix of both of those, in order to fit everything in."

The servo is controlled using the pigpio library directly from a GPIO pin. Both servo and display draw less than 500 mA, and are powered from the same USB connection so they can be powered from the Pico, with no extra power source needed.

Since it was 3D-printed, James was able to experiment with a few iterations before settling on a design in which everything fits comfortably in place. Even so, he says, the control board for the display ended up at a bit of an odd angle.

Quick FACTS

- ▶ VK-Pocket uses OpenCV to detect eyes
- ▶ And displays your eyes on a CRT screen
- ▶ The screen is recycled from an old camcorder
- ▶ The case is 3D-printed
- ▶ The video displays at 15 frames per second

OpenCV tracks facial landmarks while picamera, running on Raspberry Pi Pico, provides a 15 fps video feed

The VK machine finds the viewer's eye and displays an eerily disembodied view on an old video camera display

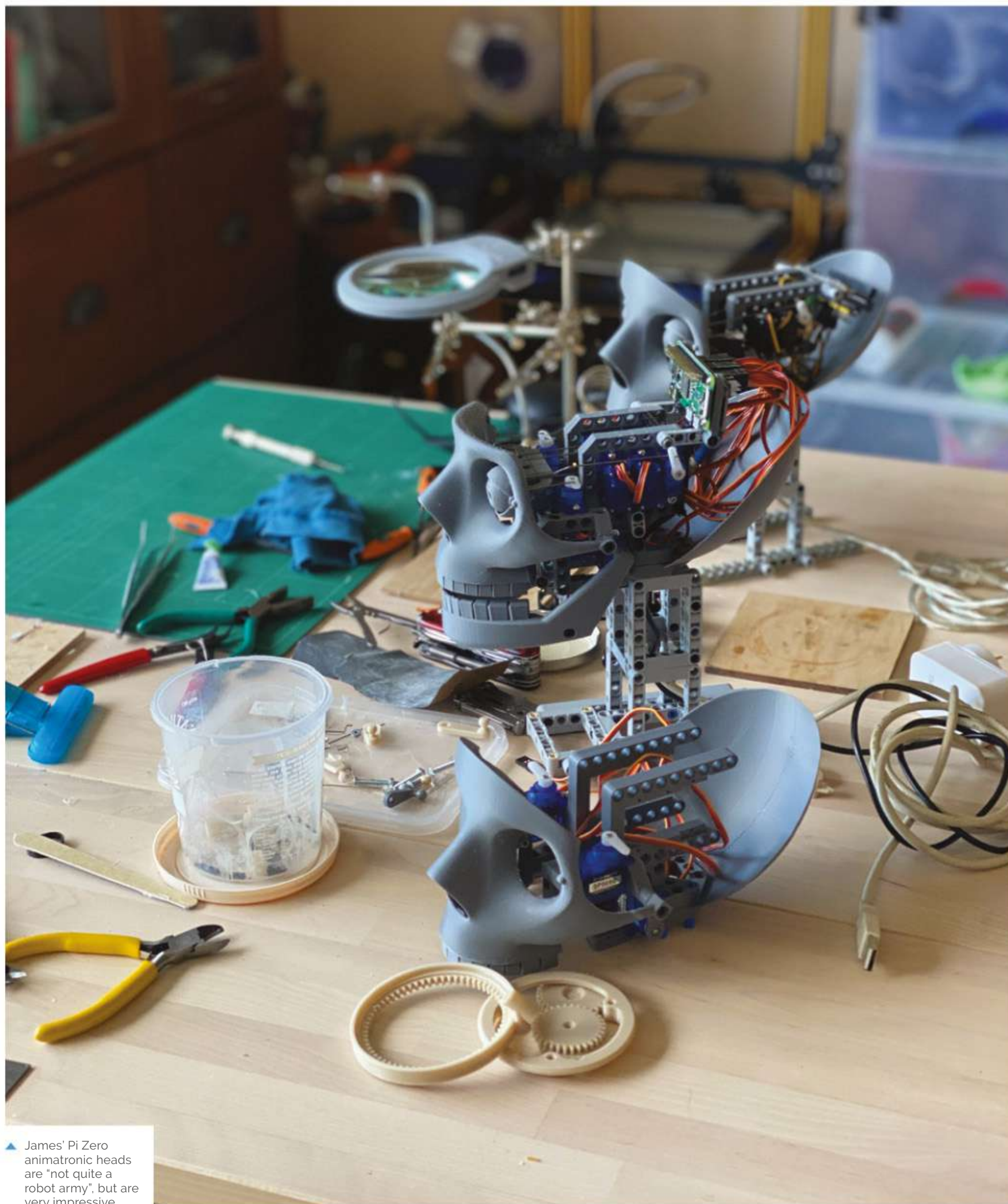
Putting the camera on a stalk turned out to be tricky, too, “so I put it inside the main case, looking out through a hole.”

The eyes have it

James wrote “a quite minimal” amount of Python code (magpi.cc/pieyepy) “to keep the high-res live video updated via the GPU while the CPU does the eye tracking.” He used OpenCV to detect faces with five facial ‘landmarks’, from which eye locations are taken. Although the eye-tracker appears to work in real-time, James realised it would be sufficient to have second-by-second updates. “If you wanted to get clever, you could use motion vectors from the compression hardware to improve tracking between detections, but it seemed good



▲ The project was inspired by the tense interview scene in *Blade Runner*



▲ James' Pi Zero animatronic heads are "not quite a robot army", but are very impressive




enough just updating every second or so.” This reduces the processor overheads and works nicely on a Pi Zero.

The Pico CPU outputs 320×240 images at “maybe a couple of frames per second”, while the picamera library keeps the screen updated with the live image. “The video hardware can handle 2592×1944 at 15 fps, and crop, scale, and display

“ Raspberry Pi Pico was my first choice for this build. I love these things! They’re a full Linux PC in a microcontroller form factor ”

it without touching the CPU, James explains. As a result, the eye region is still reasonably detailed, even though it’s only a tiny portion of the camera’s view. “If you sit still, it locks on to your eye in less than a second, and stays well centred.”

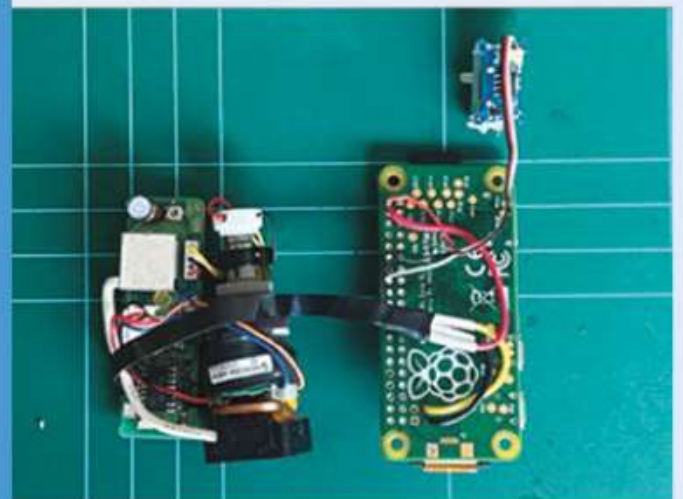
There’s no word yet from James on whether his VK-Pocket machine actively analyses its subjects’ eyes to check whether or not they may actually be a replicant. 

▲ James’ skull has eyes with embedded cameras that swivel and stare unnervingly at their prey

Go eyeball to eyeball



01 To recreate the Voight-Kampff machine, you’ll need a Raspberry Pi Pico, HQ camera, a suitable enclosure, and the CRT display from a camcorder. James designed and 3D-printed a case for his.



02 A servo drives mini bellows, like those in the original film. The servo is controlled using the pigpio library directly from a GPIO pin. No extra hardware is needed.



03 Download and install the eye-tracking code from James’ GitHub at magpi.cc/pieyepy. This will seek out a face, analyse the image, and crop into the eye area, for display on the CRT.

Psion 5MX to Raspberry Pi Zero

Kian Ryan has breathed new life into an already well-used Psion PDA and turned it into a Linux terminal, as **David Crookes** explains



Kian Ryan

Kian tinkers with old computers and tries to build new things. Sometimes at the same time.

kianryan.co.uk

It can't have escaped anyone's notice that *The MagPi* loves to see makers tinkering with retro tech. Bringing old machines back to life is humongous fun, allowing us to tap into nostalgia while seeing how far devices of a bygone age can be pushed in the modern world.

With Kian Ryan's latest project, a Psion 5MX personal digital assistant, popular in the late 1990s, has been turned into a portable Linux terminal, thanks to the addition of a Raspberry Pi Zero computer. The idea grew off the back of another project.

"I was attempting to connect a RC2014 – a Z80-based homebrew computer – to the Psion

and was having some difficulty", Kian says. "But I could get Raspberry Pi to talk to the RC2014 and I could get a Psion to talk to Raspberry Pi, so I decided to put Raspberry Pi in the middle. It worked a treat and I realised I could use Raspberry Pi for a range of other little jobs."

Write on

Kian loves his Psion. "It's still an aspirational device. The combination of screen and keyboard haven't really been beaten in the 23 years since it was first manufactured," he says. "It's a very usable, very geeky device that's near-perfect for the keyboard-orientated.

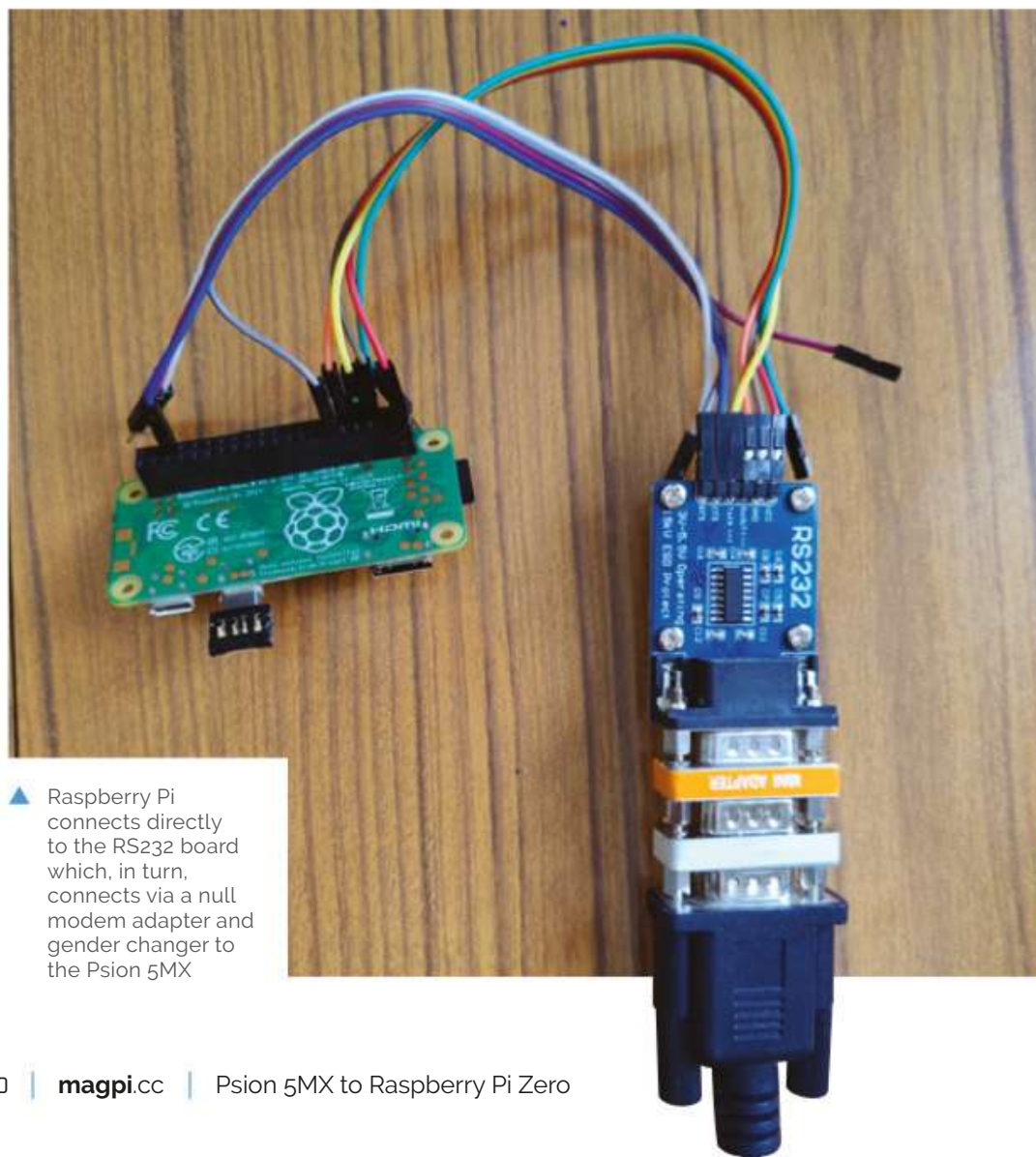
"I still use it as a daily writing driver and a good number of my blog posts start off life on the Psion before being transferred to the big machine. I wanted to give it a new lease of life to do a range of jobs that it can't currently do, which is write code, browse the internet, and occasionally tweet."

Cue Raspberry Pi Zero, a device perfect for the task in hand. "It's a nice platform to work from because it's small, light, and low-power," Kian says. "I was only looking for lightweight jobs in a console, and I didn't need the full power of Raspberry Pi 4."

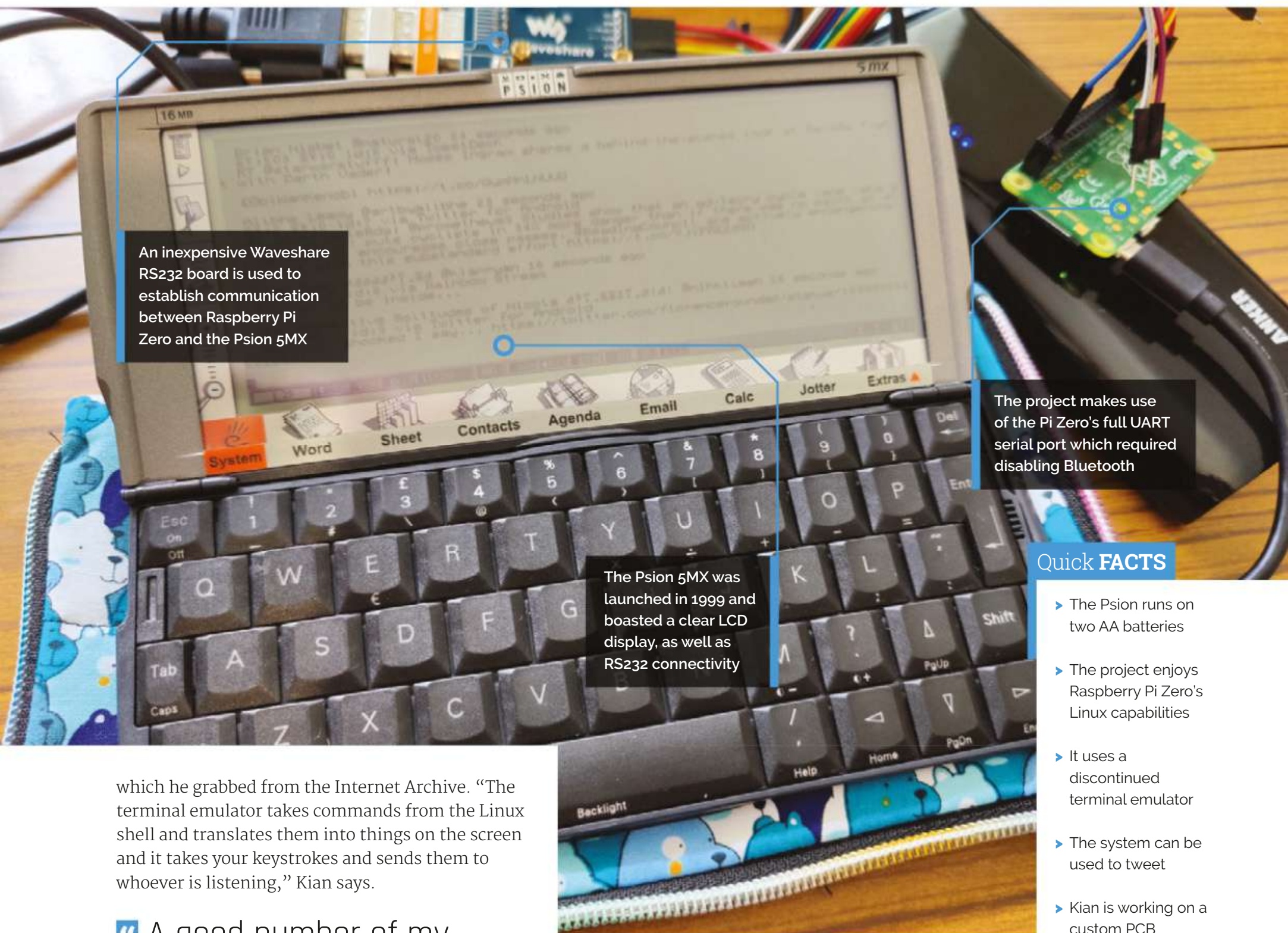
Good to talk

The idea was to use a serial connection to allow the two devices to communicate. As Kian says, it required a pile of cables and adapters. "I used a proprietary Psion RS232 to DB9, a gender changer, a null modem, RS232-TTL, and jumper leads to Raspberry Pi," he says. "I then enable the serial console through config.txt, switch the UART, enable CTS/RTS, and tell the serial driver to use hardware flow control"

It proved to be a rather involved process, with the additional fun of setting up the software. Kian has used the Hermes terminal program for Psion 5MX



▲ Raspberry Pi connects directly to the RS232 board which, in turn, connects via a null modem adapter and gender changer to the Psion 5MX



which he grabbed from the Internet Archive. “The terminal emulator takes commands from the Linux shell and translates them into things on the screen and it takes your keystrokes and sends them to whoever is listening,” Kian says.

“A good number of my blog posts start off life on the Psion”

“Using a terminal emulator and a few wires means we can take commands from Raspberry Pi and interact with them on the Psion.” It also meant Kian has been able to use a terminal-based Twitter client called RainbowStream to send a tweet from the Psion 5MX. He can also make use of the VIM text editor. “What more does a user need apart from Raspberry Pi and a mighty VIM?” he asks.

Kian is certainly happy with the result. “What I wanted from the Raspberry Pi side was something that let me fiddle,” he continues. “I’ve used it to browse Twitter a little bit and, with hardware flow control, Hermes runs VIM pretty great. It’s a pleasant experience, and fun little Linux machine.”



Talkative Tube Dashboard

Need to know if the Northern Line is OK? One maker needs to for his job, so he made it easier for himself. **Rob Zwetsloot** minds the gap



MAKER

Richard Kirby

A test manager at a company that is working to automate London Underground's railways.

magpi.cc/tubedash

There's been a huge project, going back several years, to automate part of the London Underground's railways. As you might expect, a system like this is very complex and very large, so things can go wrong. Richard Kirby, a test manager for the work, created a way to monitor the status of the network.

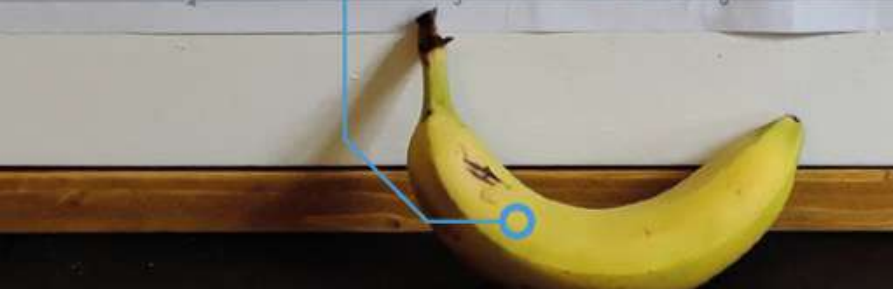
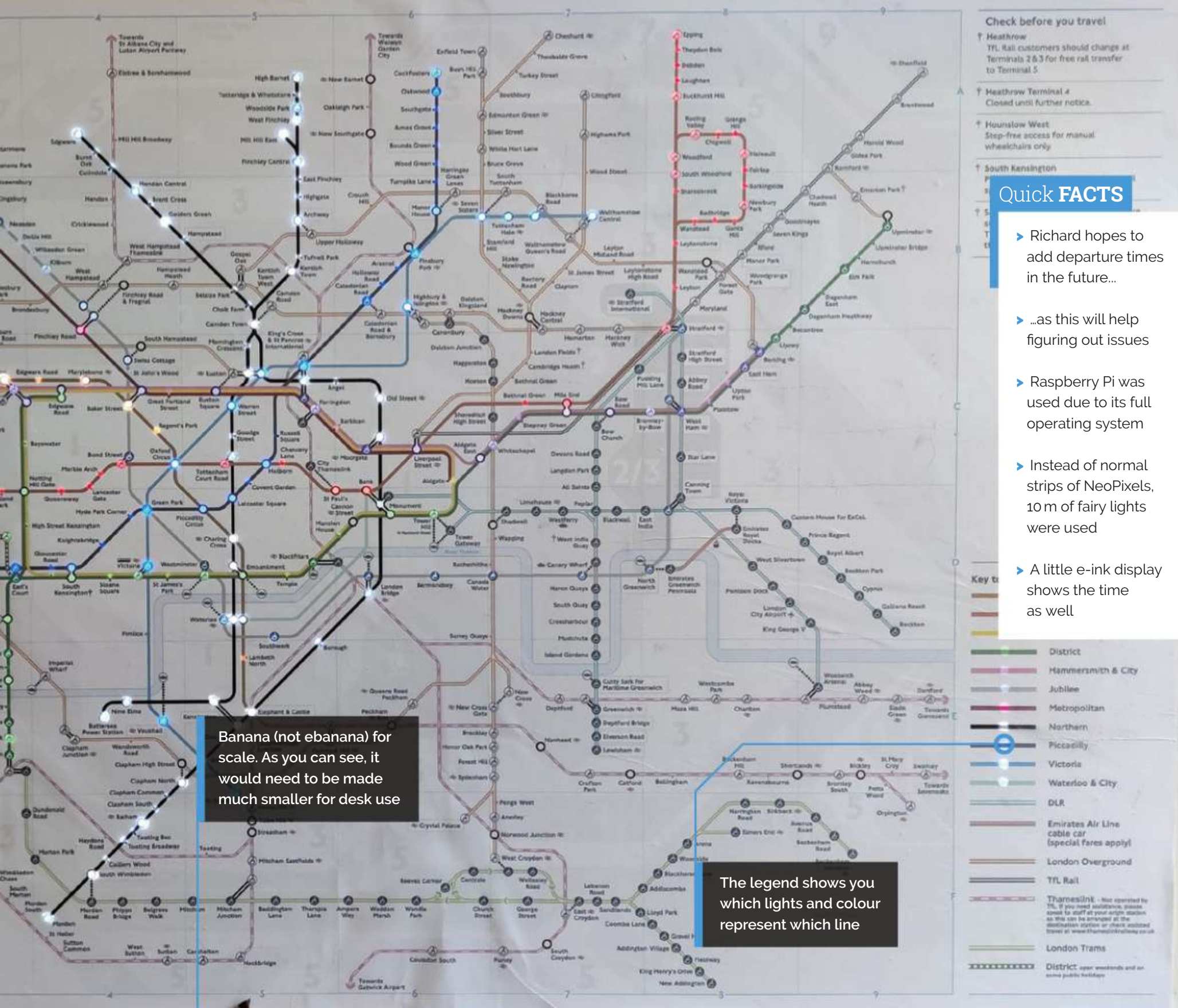
"My project is a large Transport for London (TfL) dashboard that provides the status of each line from the TfL Open Data site," Richard tells us. "It regularly queries the site and updates each line's state. A line's state is depicted by individual NeoPixel multicolour LEDs which shine through a printed TfL map via an individually drilled hole (284 holes total) in a 5 mm-thick piece of plywood. Each state results in a different flashing pattern for the line's pixels. 'Good Service' is represented by the LEDs for that line are fully lit, with no flashing; 'Minor Delays' is rapid flashing from 50% to 100% brightness; 'Severe Delays' has slower flashing, etc. The state of each line is also shown in the lower right-hand corner map legend, where the line colours are listed. I can therefore immediately see the state of all the lines at a glance, and also notice if something changes. The map also connects to a Bluetooth speaker to provide the details behind any problems, which is also from TfL Open Data."



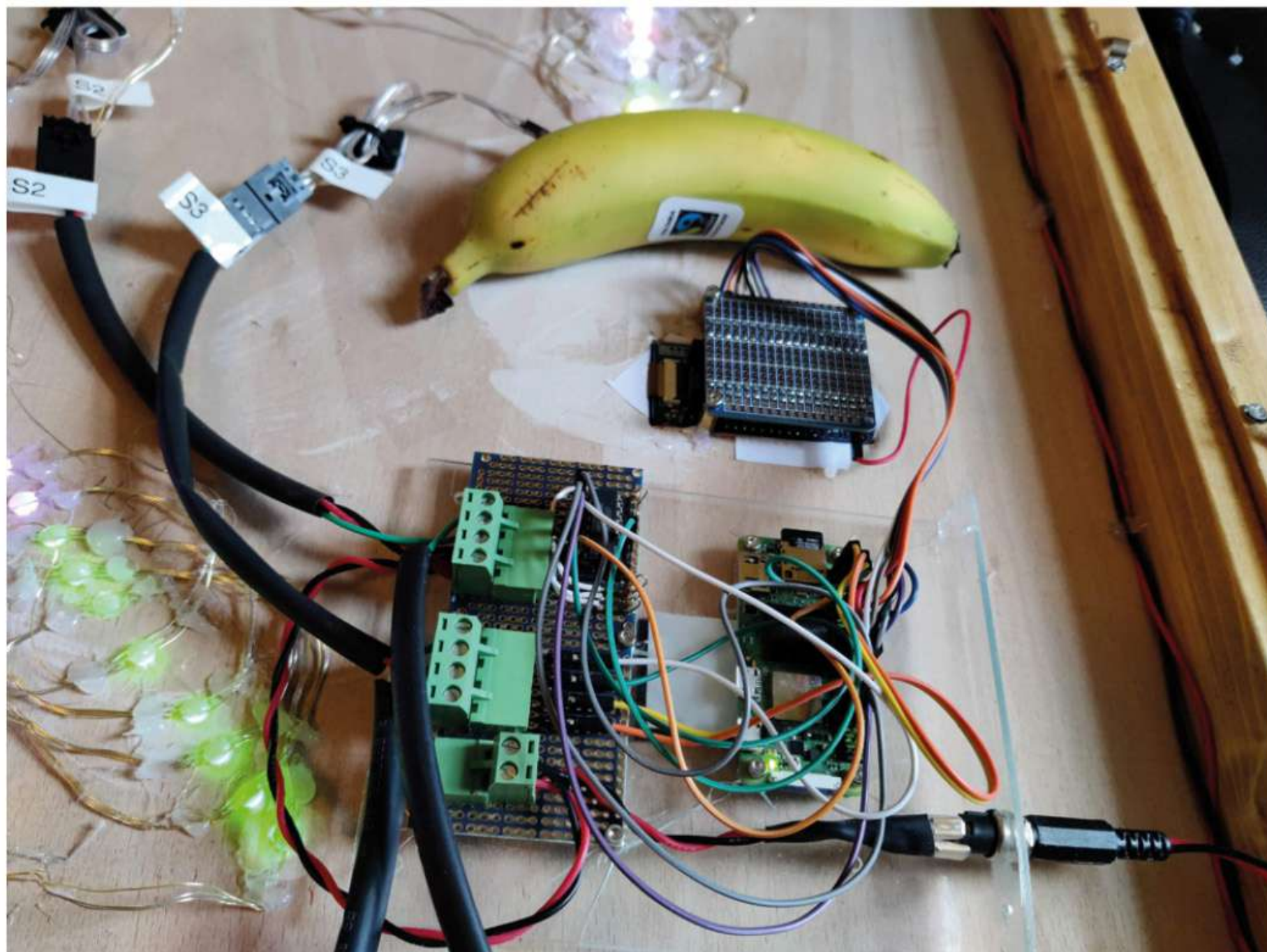
Since building the dashboard, a few system failures have been caught by Richard with it.

Light Tubes

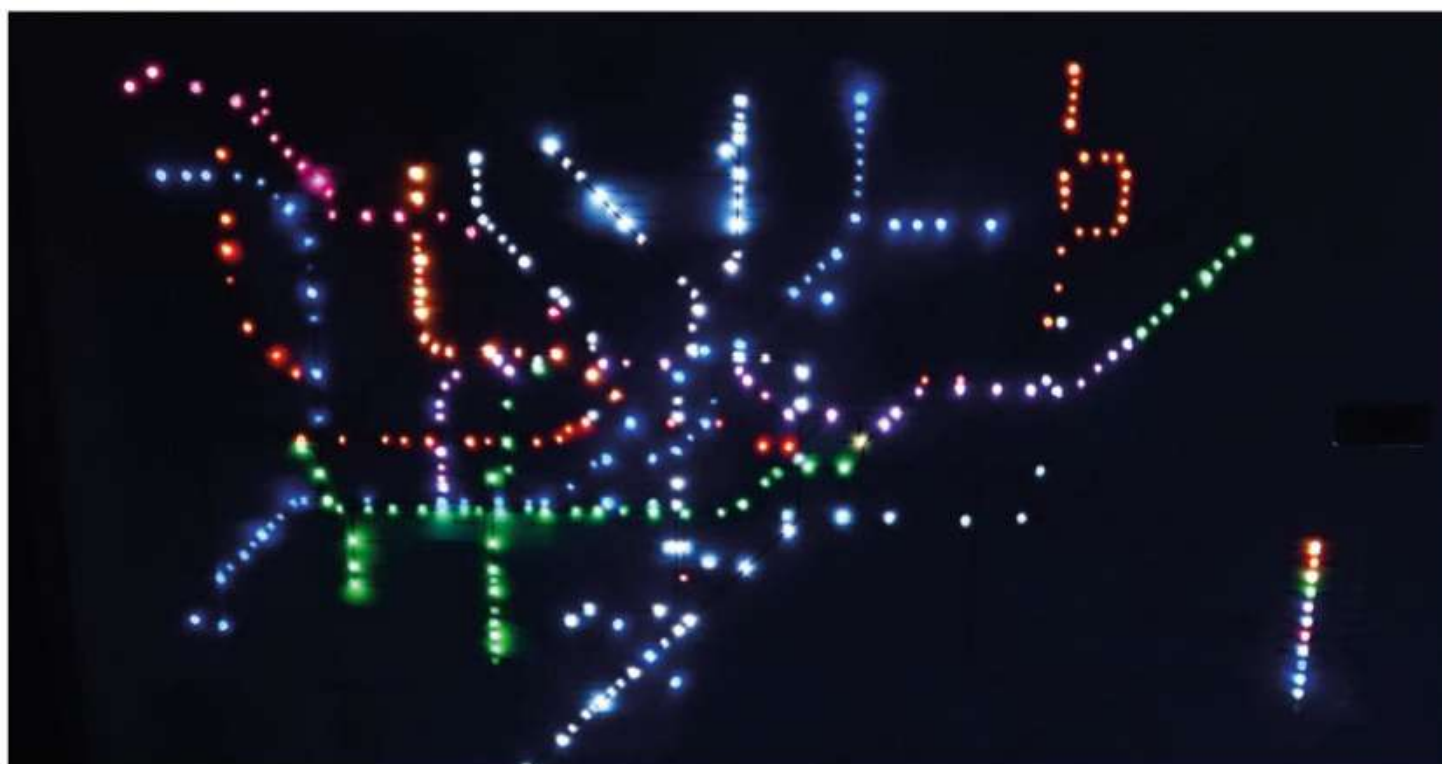
"Like all my projects, it was built in fits and starts," Richard reveals. "I built two prototypes before settling on a design. I tried electroluminescent (EL) wire first, but it was fiddly and emitted a high-pitch whine. The inverter also melted at one point, causing an awful smell."



“ My work colleagues are quite impressed as well. They can see the utility of it. One suggested putting it in the office ”



- ▲ A close-up of Raspberry Pi and the multiplexers. They're very neatly arranged
- ▶ The dashboard looks amazing in the dark, truly showing off the lights and effects



Richard is quite pleased that the dashboard came out very well without the use of 3D printers, even if it took a while to complete.

“The building of the final project took ages, as I had to continually deal with the normal problems, wait for tools, etc.,” he admits. “I only work on Raspberry Pi projects occasionally, so a lot of time elapses from beginning to end. I had to buy a router to make the frame – it is intended to be hung on the wall. I haven’t gotten around to hanging it yet.”

A Raspberry Zero 2 W is the heart of the project, querying TfL Open Data to get the statuses of the various lines, which it then converts into the animations listed above for the four different NeoPixel strings with 100 RGB LEDs each.


“Raspberry Pi also makes service announcements if there are any major issues... this is done via the text-to-speech engine at Google Translate, and a Bluetooth speaker.”

Signal status

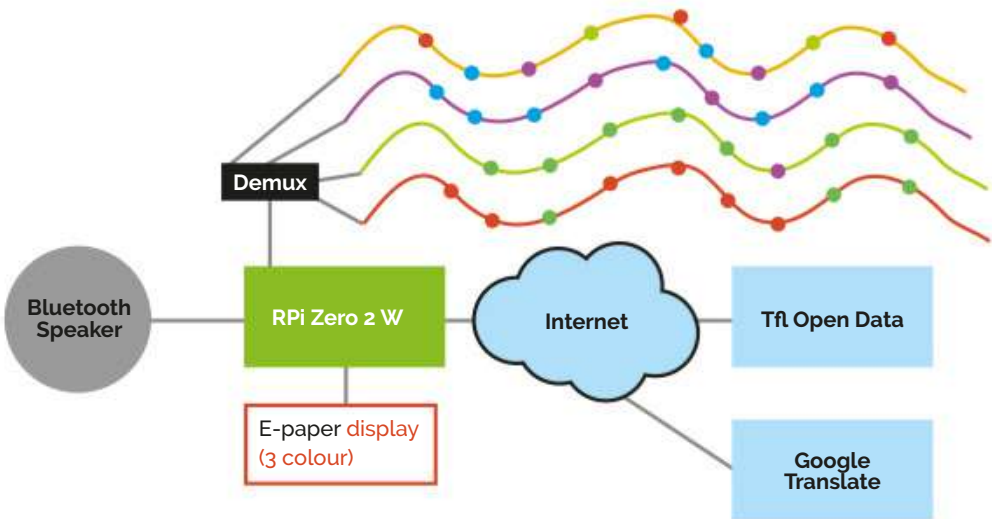
You can see the dashboard in action from a talk at Raspberry Pint: magpi.cc/dashvideo. People there liked it, and so did his family at home.

“My wife is very impressed – she likes the concept and it is relatively attractive compared to other things I have built,” Richard says. “Her mother, my mother-in-law, was also impressed and was happy to provide some advice on how to glue the paper to the board – she is Japanese and does Japanese calligraphy, so was able to provide excellent advice. I didn’t do a very good job, but it would have been worse without her advice.”

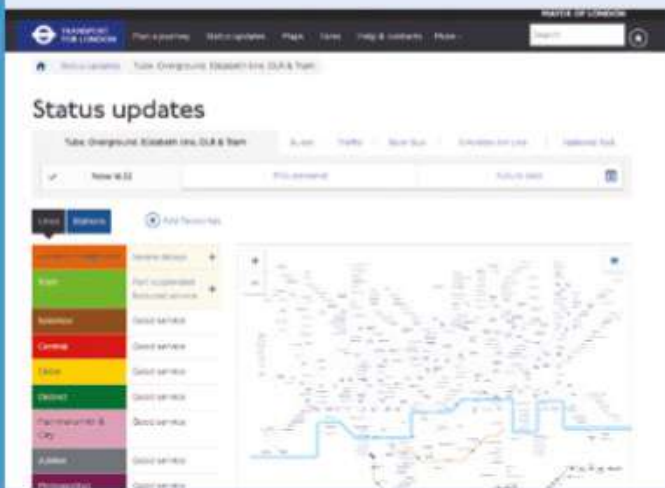
Richard and his family also use it to plan their trips into central London regularly – at a glance able to know which lines are having problems.

“My work colleagues are quite impressed as well. They can see the utility of it. One suggested putting it in the office – I might make a small version for my work desk.” 

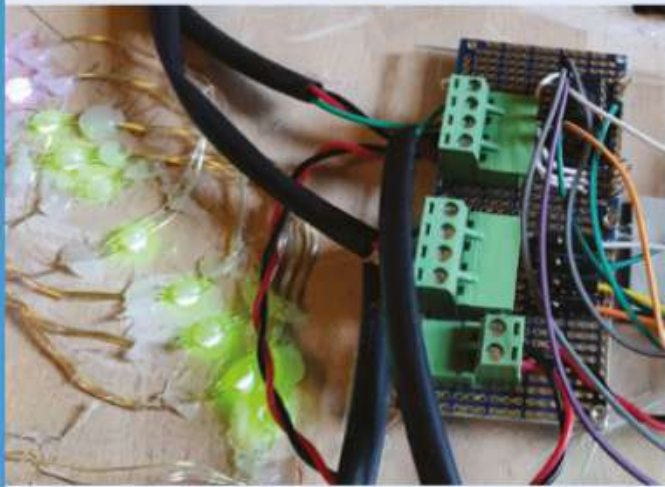
▼ This is a rough flow chart of the entire system, based around Raspberry Pi Zero 2 W



Detecting a fault



01 When a problem occurs on the London Underground, the issues are made available so that staff, travellers, commuters, etc. know the state of the services. This data is available via the TfL API, which is captured by the dashboard.



02 The dashboard reads the API to set the NeoPixels: “The demultiplexer is needed to drive the four strings off of a single pin (pin 18). This was necessary as they are fixed-address NeoPixels The TfL open data site has a fairly simple REST interface, which supplies statuses as well as arrival/departure information.”



03 The LEDs are lit up in a specific manner to represent different statuses. “It enables me to quickly know there was a problem, how long the problem lasted, and how serious it was. In one case, I was able to arrange some help to be dispatched... following a failure.”

CogniFly

This Raspberry Pi Zero W-powered quadcopter is smart, ultra-light, and nigh on crash-proof, as **Rob Zwetsloot** discovers



Ricardo de Azambuja

An electrical engineer with a PhD in Computer Science, Robotics, and Artificial Intelligence who describes himself as a Mad Scientist. He works as a research associate at MISTLab.ca.

magpi.cc/cognifly



Warning!
Drone Safety

Be careful when working with drones, especially those with spinning blades. Fly safely and responsibly and follow local laws and regulations.

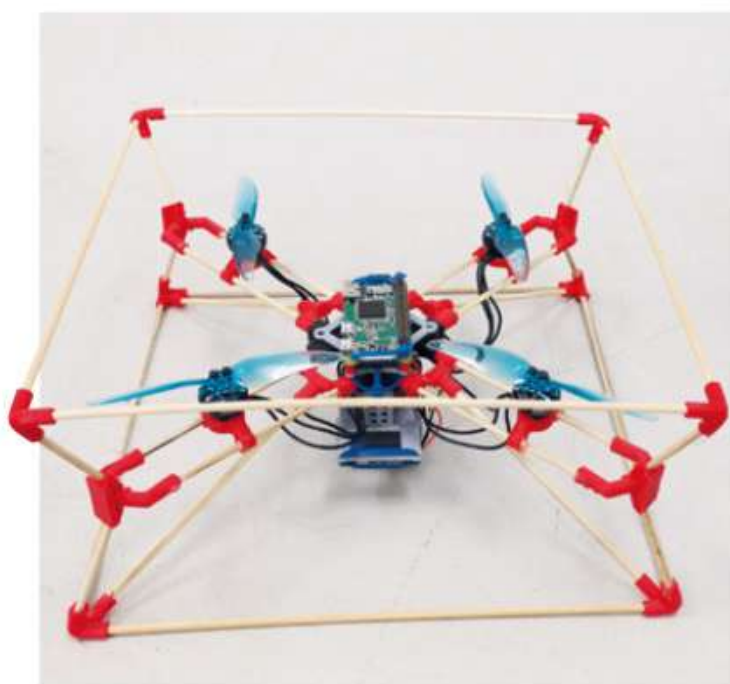
magpi.cc/dronesafety

► Bamboo is also a perfect building material for CogniFly

Gravity may not be the strongest force in the universe, but it is something that we contend with on a daily basis. When you're trying to get off and stay off the ground, though, it gets even trickier. Weight is incredibly important, and usually that means you can do less in the air. Ricardo de Azambuja found a way around it for his quadcopter.

"The CogniFly is an open-source, collision-resilient, under-250g quadcopter that is easy to build and to program," Ricardo tells us. "But don't let its light weight make you believe it lacks computational power: thanks to the Raspberry Pi Zero W, it can do amazing things with a Raspberry Pi Camera Module, and the use of an external neural network accelerator (Google AIY HAT or Coral Edge TPU) gives it superpowers!"

While solving this problem, Ricardo also managed to work on another – toughness. Ricardo reckons it needs very little maintenance, even if it is crashing around the lab a lot. With this confidence, a swarm of clones of them are being made at MISTLab.ca.



and that was not different for the CogniFly," Ricardo says. His postdoctoral project proposal was to create a small, autonomous drone for 'precision agriculture'. Looking at small, commercial quadcopters, his team realised the closed source nature and external computer dependency didn't fit with the project, which is why they went with Raspberry Pi.

"From that point, we decided to build our own drone and that was when we realised how important collision resilience was!" Ricardo explains. "As we were trying to test and debug our initial designs, their 3D-printed frames would crack, or the propellers would be smashed into

“ I already knew other interesting works that explored insect-inspired robots ”

pieces. During my PhD, I was exposed to many ideas from collaborative robotics, soft-robotics and morphological computations (e.g. the GummiArm, BAXTER), therefore it was easy for me to transfer those experiences into flying robots. I already knew other interesting works (e.g. from Imperial College, EPFL, University of Pennsylvania, and ETH Zurich) that explored insect-inspired robots, so, after watching some videos about bees crash-landing, and reading more on how insects managed to survive that... I had the idea of making a quadcopter that shouldn't work according to popular belief because its motors and the flight controller (IMU) were not rigidly connected."

The result of that research is a frame using bamboo rods or carbon fibre, and other small 3D-printed parts.



The rotors last a lot longer than in other drones, and rarely get hit, unless there's a huge collision

A lot of calculations are done on board instead of on an external computer

Carbon fibre rods are lightweight and flexible. Paired with the joints, it allows for a lot of collision resistance

Quick FACTS

- ▶ It uses the camera and optical flow in parts of its automation
- ▶ It can also be controlled using a PS4 controller
- ▶ The flexible joints can absorb smaller collisions...
- ▶ ...while the whole frame will (safely) deform on big hits
- ▶ A Raspberry Pi 3A+ was considered but was too heavy

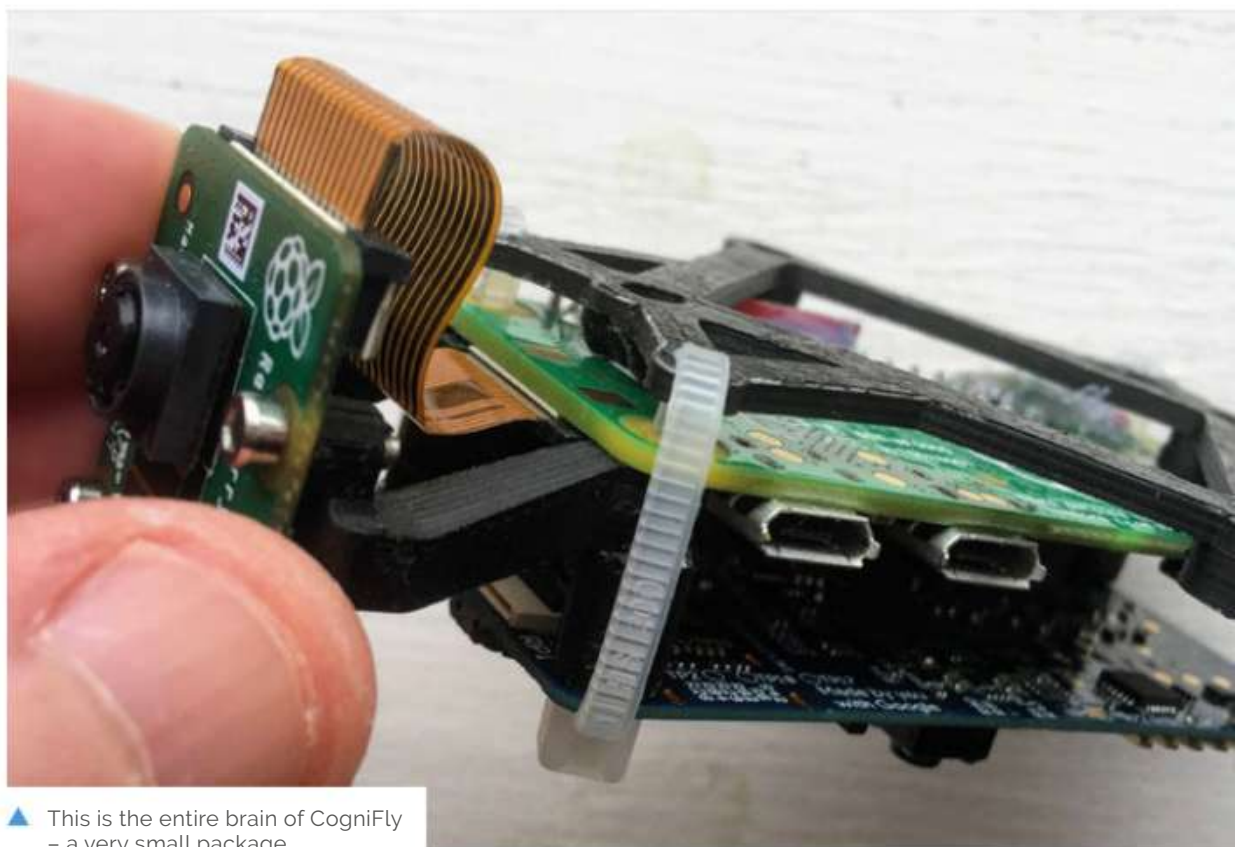
Fly it yourself

CogniFly was recently shown off at ICRA 2022, a robotics and automation conference.

“The reaction was amazing,” Ricardo mentions. “Some people even suggested the CogniFly was the most interesting robot they saw at the conference that year. Before that, we received some nice feedback from the community and from **AviationConnection.org**, where secondary students were using our drone to learn about artificial intelligence.”

Ricardo believes that it's very easy for everyone to build one themselves – all they need is access to a 3D printer and some minor soldering skills.

“Our plan is to keep developing it, hardware and software – and, as it's open-source, we hope the community will come up with new improvements and different applications for the drone. In relation to the single-board computer, we are currently exploring Raspberry Pi Zero 2 W and the Google Coral Dev Board Mini.”



▲ This is the entire brain of CogniFly – a very small package

SUCCESS STORY magpi.cc/success

Pimoroni PicoSystem

Rising component costs helped push Pimoroni towards developing a hero product on a yet-to-be-tested Raspberry Pi platform.

Co-founder Paul Beech explains why. By **Rosie Hattersley**

Sheffield-based Pimoroni is a hobbyist store where electronics enthusiasts find the full complement of Raspberry Pi products and associated accessories, as well as compatible Raspberry Pi HATs and hardware to augment the Raspberry Pi project-building experience.

The 40-strong Pimoroni team is staffed by techies who love coming up with new ways to make use of coding, and enthusing others about the possibilities of DIY computing. Their products are designed and manufactured in-house, and stocked by more than 100 distributors worldwide.

The company was founded in 2012 by digital designer Paul Beech and Jonathan Williamson, a “precision-junkie” and ninja coder. Having designed the Pibow – a custom case for the then

recently-launched Raspberry Pi – using just a laser cutter and a kettle, it proved so successful that they decided to create a business as a ‘purveyor of maker goods’ to serve the emerging DIY electronics and computing market. The single board computer remains Pimoroni’s biggest seller, although it also stocks Arduino, micro:bit, and wearable computing hardware.

THE CHALLENGE

Pimoroni’s success is down to the rapid development of products for engineers, makers, and educators,” says director and co-founder, Paul Beech. Many of their products come about as a direct result of requests from customers, and they have built up a deserved reputation of developing and stocking the right products at a fair price point, and with consistent availability. Being on good terms with Raspberry Pi from the outset has been immensely helpful for both sides.

When supplies of a key component in one of their gaming products began to run low, Pimoroni decided to find a more cost-effective replacement. “We had the dual challenge of an existing product which used a chip with high-cost and poor availability.” They were concerned about being unable to continue a product line, while rising prices for those they could secure lead to poor market competitiveness. “The key part we had problems with had a list price around \$5. During shortages,



this leapt to over \$20 and availability was still poor,” says Paul.

THE SOLUTION

Pimoroni’s existing relationship with Raspberry Pi meant they were privy to plans for a brand-new microcontroller, the RP2040. They decided to press ahead with developing a handheld gaming console in the maker/consumer space based on it. Paul says RP2040 gave much of the same experience at a fraction of the cost [of the previous component they were having trouble sourcing]. They were also swayed by the ready availability, technical capabilities, and the excellent documentation and examples to accompany the RP2040 microcontroller.

Pimoroni built “an entire gaming API” around the RP2040, using its unique PIO [programmable input/output] features “to achieve great display and audio performance with small MCU overheads.”

“We now use RP2040 in around 80% of our products that have a microcontroller”

Far from having to compromise on the capabilities of their reimagined mini gaming board, “we’re running the full gamut of hardware, such as audio output, displays, accelerometers, LiPO/battery circuits, and various sensors,” he says. There’s now a community-supported ecosystem around the API.

WHY RASPBERRY PI

Raspberry Pi’s documentation and community made it easy to just get on with development. “If any really obscure issues appeared, we knew



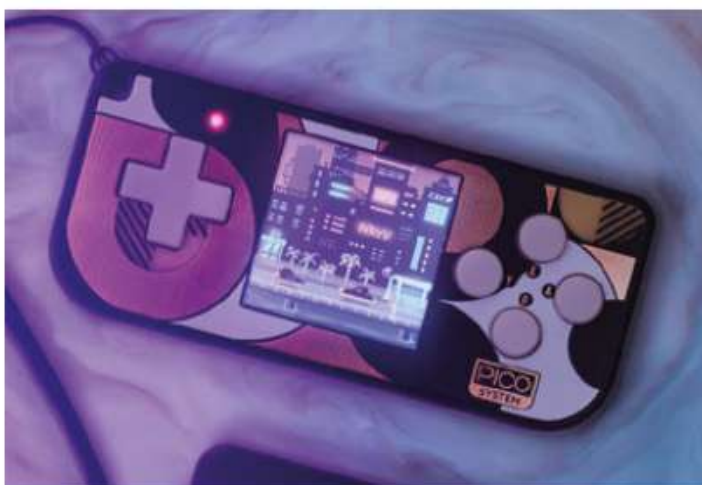
Raspberry Pi’s engineers would be there to help but, so far, there hasn’t been a need,” says Paul. “The RP2040 offers exceptional capabilities for the price and much better availability, alongside stellar documentation that makes rapid development a low-stress proposition.”

THE RESULTS

The PicoSystem gaming console – described by Pimoroni as ‘a tiny, complete gaming handheld made from layers of step-milled PCBs and love’ – launched alongside the RP2040 itself in January 2021. It’s been enthusiastically received by gamers, some of whom have gone as far as to develop mini games for it on the indie platform itch.io.

“We’ve been low-key, waiting for Eben and Raspberry Pi to do something with microcontrollers for years, and we’re not disappointed in the RP2040 and the support around it. As soon as we got word, we went full-on to make the best of what we do, but for this new microcontroller,” Paul explains on the Pimoroni blog. “The documentation and examples for RP2040 were a real step-up, not just for Raspberry Pi, but for the microcontroller industry. It felt more in touch with modern software and firmware development,” he tells us.

A year on, Paul says the manufacturing yields for RP2040 have been better than the original part. “This is a dual function of a simpler, yet capable, design and an easier part to work with. This is despite our routine use of the RP2040 at twice its official clock speed. We now use RP2040 in around 80% of our products that have a microcontroller, as there is very little rationale for using any other part.”

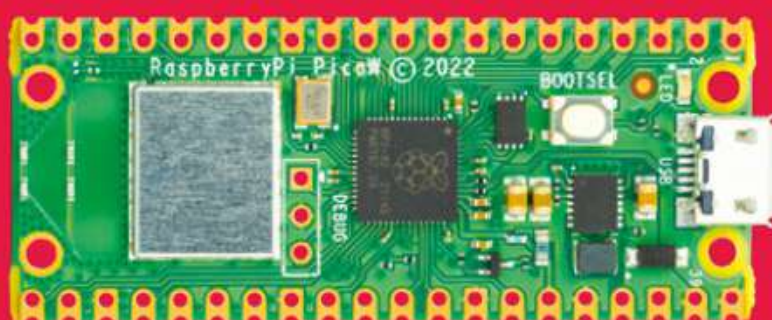


SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W



+



Subscriber Benefits

- ▶ **FREE Delivery**
Get it fast and for FREE
- ▶ **Exclusive Offers**
Great gifts, offers, and discounts
- ▶ **Great Savings**
Save up to 35% compared to stores

Subscribe for £10

- ▶ Free Pico W
- ▶ 3 issues of The MagPi
- ▶ Free delivery to your door
- ▶ £10 (UK only)

Subscribe for 12 Months

- ▶ Free Pico W
 - ▶ 12 issues of The MagPi
 - ▶ Free delivery to your door
- | | |
|----------|---------------------|
| £55 (UK) | £90 (USA) |
| £80 (EU) | £90 (Rest of World) |

☎ Subscribe by phone: **01293 312193**

📧 Subscribe online: **magpi.cc/subscribe**

✉ Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £30 every six months unless cancelled. A free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. (No Raspberry Pi Pico W with £5 Rolling Monthly Subscription).

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print
today and get a **FREE**
development board

- ▶ A brand new
RP2040-based
Raspberry Pi Pico W
development board
- ▶ Learn to code with
electronics and build
your own projects
- ▶ Make your own
home automation
projects, handheld
consoles, tiny
robots, and much,
much more

WORTH
\$6



This is a limited
offer. Not included
with renewals. Offer
subject to change or
withdrawal at any time.



Buy now: magpi.cc/subscribe

SUBSCRIBE
on app stores

From **£2.29**

Available on the
App Store

GET IT ON
Google Play

SUPER SIMPLE ROBOTICS

A first foray into the exciting world of robots needn't be scary. There's plenty of help along the way, says **Rosie Hattersley**

It's nigh on impossible to ignore the rise of the robot; they're everywhere these days: delivering groceries, harvesting our crops, assembling cars, and packaging goods in factories, or entertaining our kids.

Robots are certainly an entertaining form of toy, with plenty of educational value, not least because robots can be taught to learn from us, or the objects we show them, and to react accordingly.

Learning how to code and control a robot is one of the great rewards and challenges of being a Raspberry Pi owner. If you're fairly new to Raspberry Pi, and haven't had much opportunity to experiment with robots or with coding, taking your first step into the world of robots can seem rather daunting. However, there are plenty of kits out there to help you test the waters. Both Raspberry Pi, and the companies that create robot kits, provide step-by-step guidance and support.

“ We'd love to hear where your robot adventures take you ”

Once you've learned what's involved, and the components you'll need to create a robot of a certain type, you may even feel inspired to design a robot of your own, thus opening the door to a whole new world of creativity and learning! Our tutorial will take you through the excellent CamJam EduKit #3, providing a detailed guide to starting your robot-building journey. We promise that you won't regret giving it a go, and we'd love to hear where your robot adventures take you.



1 WHEELS

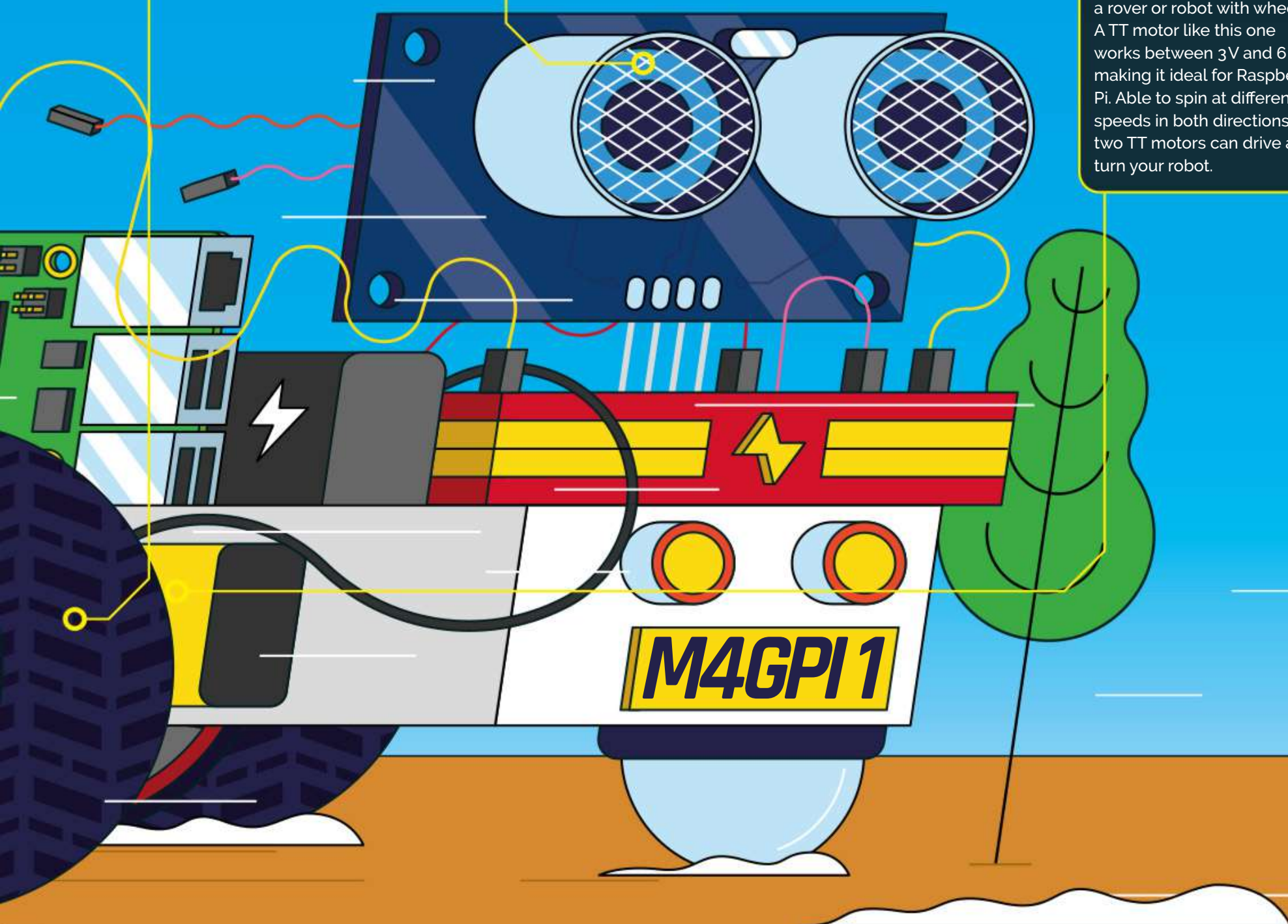
Two wheels good, four wheels bad? Or, do you want a super-maneuvrable castor so your robot can spin and accelerate in any direction? Robots and rovers come in many forms

2 DISTANCE SENSOR

Being speedy is great, but avoiding objects – and other robots – is a must. A distance sensor sends out sonar signals looking for what's in your robot's path and tells it to react or change course accordingly

3 MOTOR

You'll need a motor to drive a rover or robot with wheels. A TT motor like this one works between 3V and 6V, making it ideal for Raspberry Pi. Able to spin at different speeds in both directions, two TT motors can drive and turn your robot.



THE PARTS THAT MAKE A ROBOT MOVE

Modular parts make for a mighty robot

Designing a robot from scratch involves having a clear vision of what you'd like your creation to be able to do, and breaking down into discrete functions how it's going to achieve each stage. In many cases, you'll find robot kits available that can achieve some or all of what you'd like to accomplish and may be able to adapt to your needs once you've seen how they operate. Working through the assembly process and learning to control a commercially bought kit is useful and rewarding in its own right, and will give you the confidence to go on and design your robot.

Helpfully, many robot kits are Raspberry Pi-based (or have a Raspberry Pi component) and come with, or can be powered by, a Raspberry Pi. There are now Pico microcontroller-based robot parts, such as Kitronik's Motor Driver board for Pico (magpi.cc/picodriver).

A set of simple TT motors is all you need to get moving

Whether you are taking the plunge and designing a custom robot, or sizing up a suitable kit, these are the essential parts your robot will almost certainly feature.

Raspberry Pi

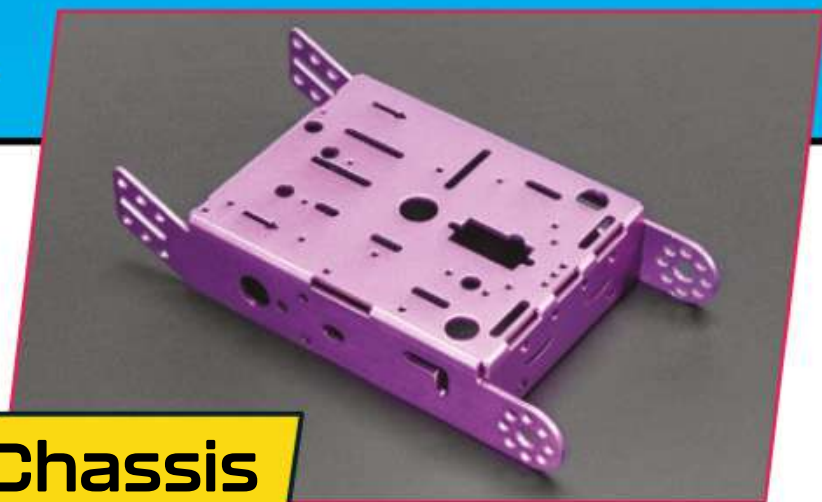
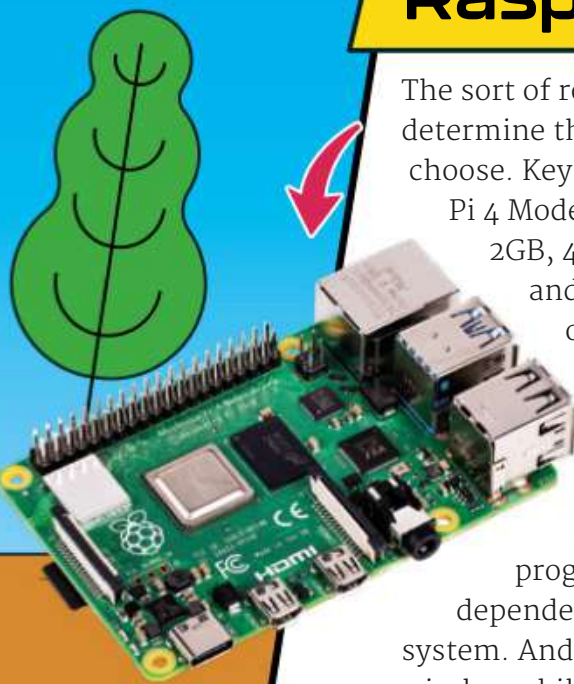
The sort of robot you want to build will determine the Raspberry Pi version you choose. Key benefits of a Raspberry Pi 4 Model B are its choice of 1GB, 2GB, 4GB, or 8GB configurations and support for AI, a full operating system, plus USB ports. The Pico W board, meanwhile, packs low power into a sleek and stealthy form factor and it powers on, and runs, programs instantly without the dependence of a larger operating system. And Pico W's new-found wireless ability makes it ideal for remote control projects.

magpi.cc/products

Chassis

Your robot will almost certainly need a body. You can either buy or build (or 3D-print) a chassis. The Pi Hut sells this striking purple chassis, which makes a great body for a two- or four-wheeled robot. It's made from a single piece of 2 mm thick aluminium, and has holes and slots for servos, sensors, and mounts. Extra holes can be drilled if required. The precision-engineered robot part is crafted to fit a pair of DC TT motors.

magpi.cc/purplechassis



Motor Controller

Motor controllers offload the speed and direction of your wheels from Raspberry Pi, making it much easier to control movement with code.

For example, this ThunderBorg dual motor controller provides 5 amps of power to two separate motors, making it a great choice for robots that require

plenty of oomph! That power is adjustable too, giving you great control over the robot driving or operating experience, including switching into reverse. ThunderBorg is stackable and can be used in multiples for the ultimate in robot power: the MonsterBorg kit gives you an insight into what this controller can do.

magpi.cc/thunderborg

Ultrasonic Distance Sensor

How does your robot or rover work out when there's an object it needs to avoid? Using a distance sensor, of course. This one uses twin speaker-like ultrasound transmitters and a receiver to detect obstacles and work out how far away they are. Your code then cleverly instructs the robot to take avoiding action.

magpi.cc/hrso4

Robot Resources

Make sure that you check out these resources

LEARN ROBOTICS PROGRAMMING 2ND EDITION

Getting to grips with coding in Python will help you get the most from your Raspberry Pi, and will unlock the potential of robotics. This excellent book provides a thorough grounding in how to design and build a robot, then add AI intelligence, lights, and sensors.

► magpi.cc/learnrobotics2



ROBOTICS PROJECTS

Robots are one of the coolest ways to use Raspberry Pi – and one of the most popular! Raspberry Pi's walkthroughs show you how to build your own robot buggy and motor, get your robot to follow a line, explore its environment, and control it remotely.

► magpi.cc/robotprojects

SG90 Servo

While motors spin, servos rotate back and forth, making them ideal for robotic joints. You'll find them in arms, pinchers, and walking robots. The SG90 is a classic part that you will find in many robot kits. It rotates up to 180° (90° in each direction).

This invaluable part wields plenty of power and connects to many motor controllers via its 3-pin female header.

magpi.cc/sg90

TT Motor

This TT motor is an easy-to-use item that you connect to a breadboard via its male jumper leads, or to the aforementioned motor controller to power up and spin around. Dispensing with rotary power dials, this motor offers 200 rpm and between 3 V and 6 V of power (but be aware that it also draws 1.5 V while idling – worth considering if you're making a battery-powered robot, as it doesn't have an 'off' switch). More expensive motors provide higher speed, gear ratios, and encoders that enable precision movement. But a set of simple TT motors is all you need to get moving.



magpi.cc/ttmotor



ROBOTICS WITH RASPBERRY PI

Providing a more pedagogical approach to robotics, Udemy's online course takes you through the essentials of Python coding before using it to control a robot, access on-board cameras, and use robotics in tandem with the Internet of Things and artificial intelligence.

► magpi.cc/udemyrobotics

RASPBERRY PI ROBOTICS KITS

Try a self-contained package and build your robot confidence

The CamJam EduKit #3

CamJam's EduKit #3 is an absolute classic kit for anyone wanting to experiment with robots. The kit consists of a motor controller, two DC motors, a distance sensor, plus a ball castor for direction changes, red wheels, and all the jumper leads, connectors, and resistors that you need.



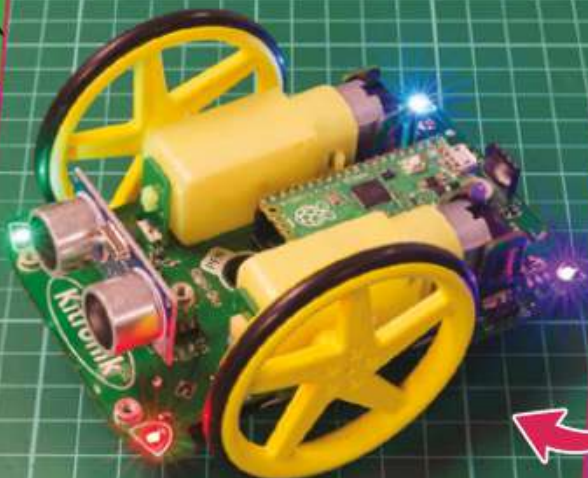
magpi.cc/edukit3

■ CamJam's EduKit #3 is an absolute classic ■

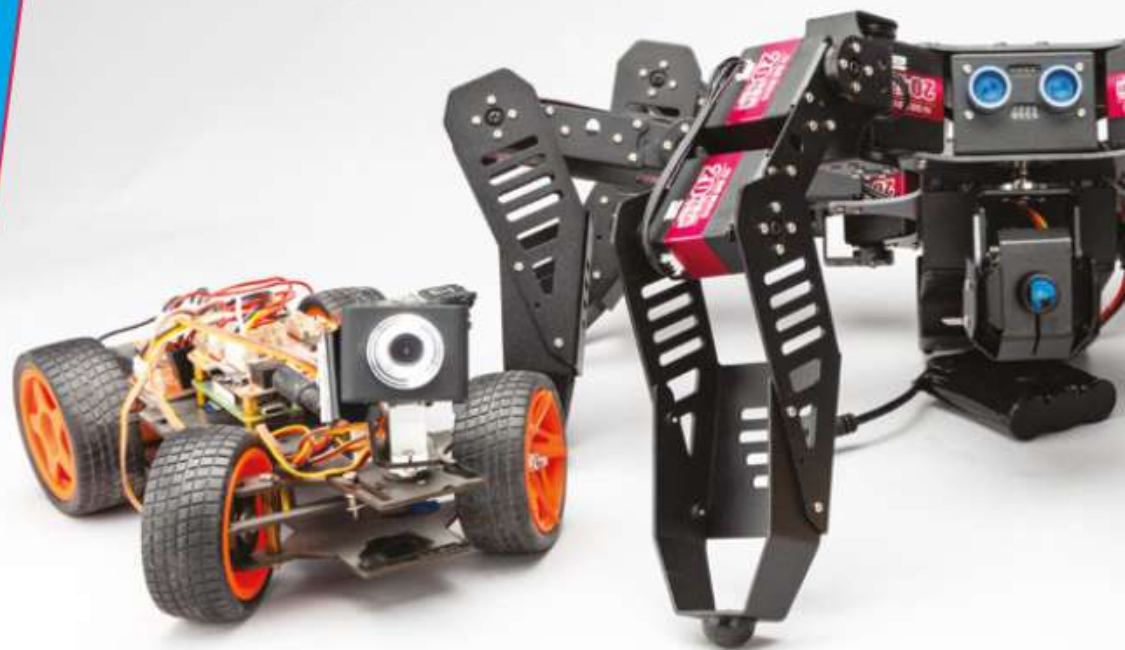
Kitronik Autonomous Robot

Nifty turns and changes of direction are in the DNA of this Kitronik buggy platform, specially designed for Pico users. With piezo buzzers, lights, and line following capabilities, basing a robot around this buggy will have you absolutely bossing it over the competition.

magpi.cc/kitronikpico



You'll
love
SpiderPi



SpiderPi

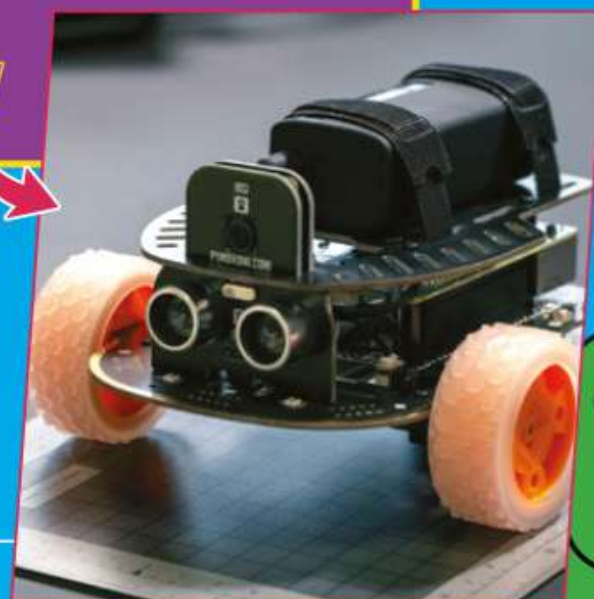
If your idea of a robot runs around and scares small animals, you'll love SpiderPi, a hulk of flexing metal that responds to visual cues and is user-programmable. It can follow lines on the ground, recognise objects, jump up and down thanks to an inverse kinematic gait, and even pick up and move objects. One of the most detailed robots around.

magpi.cc/spiderpi

Trilobot Base Kit

Sporting some serious bling, including RGB LED under-lighting, this educational but generally awesome robot kit sandwiches most of its electronics between two slices of circuit board that Pimoroni has pimped with an ultrasonic sensor, a built-in camera mount for AI cleverness, and sockets for additions such as STEMMA and Qwiic connectors. Add an SD card, power pack, and Raspberry Pi – or choose the self-contained kit.

magpi.cc/trilobot



Pi-top Robotics

If you're looking for a high-end robot kit for classroom or code club use, this robust setup is a great choice. Featuring more than 50 metal plates, rugged wheels, a camera, and an ultrasonic sensor, plus a range of servo motors, this STEM-focused setup works alongside, or can be bought with, a pi-top 4 Raspberry Pi portable computer.

magpi.cc/pitoprobot

GoPiGo 3

One of the most functional robot kits available for Raspberry Pi. GoPiGo broadcasts its own wireless hotspot, making it easy to connect and the two motors have encoders built-in, making for precision movement. One of the most precise robots you can buy, and it is especially useful for teachers.

magpi.cc/gopigo



BUILD YOUR OWN ROBOT: CAMJAM EDUKIT #3

Robots are a great way to learn physical computing and lots of fun too. Don't know where to start? This is the tutorial for you



MAKER

PJ Evans

PJ is a writer, software engineer, and general tinkerer. He believes no robot is finished until it has googly eyes.

twitter.com/mrpjevans

You'll Need

- ▶ CamJam EduKit #3 magpi.cc/edukit3
- ▶ 4 × AA batteries
- ▶ USB power bank (optional)

Raspberry Pi computers have often been the controller of choice for robot builders. Its small size, combined with the power of Raspberry Pi OS, makes it an ideal choice for simple 'buggy' projects and complex machine-learning autonomous builds alike. By combining Raspberry Pi with a battery, you can make untethered robots that can be controlled by Bluetooth, wireless LAN, or radio. There are literally thousands of options out there, so getting started can be intimidating. We've hand-picked some great beginner kits, and will walk you through them over the next few issues, starting with the CamJam EduKit #3.

01 Get to know your kit

The CamJam EduKit #3 Robotics kit contains everything you need to build your first robot. Inside, you'll find all kinds of bits and bobs. Don't be put off, we'll go through all of these



▲ You can use the battery pack and Raspberry Pi 4 as 'ballast' at the rear. Add a power bank for true remote computing!

in time. We're going to start with some basic components and then build on that. Of interest to us are the two motors (which will drive the robot), the big grippy wheels, and the castor ball, which provides the rear 'wheel'. You may be wondering what we're going to mount these on. Well, remove all the components and safely store them, because we're going to use the box itself as the chassis!

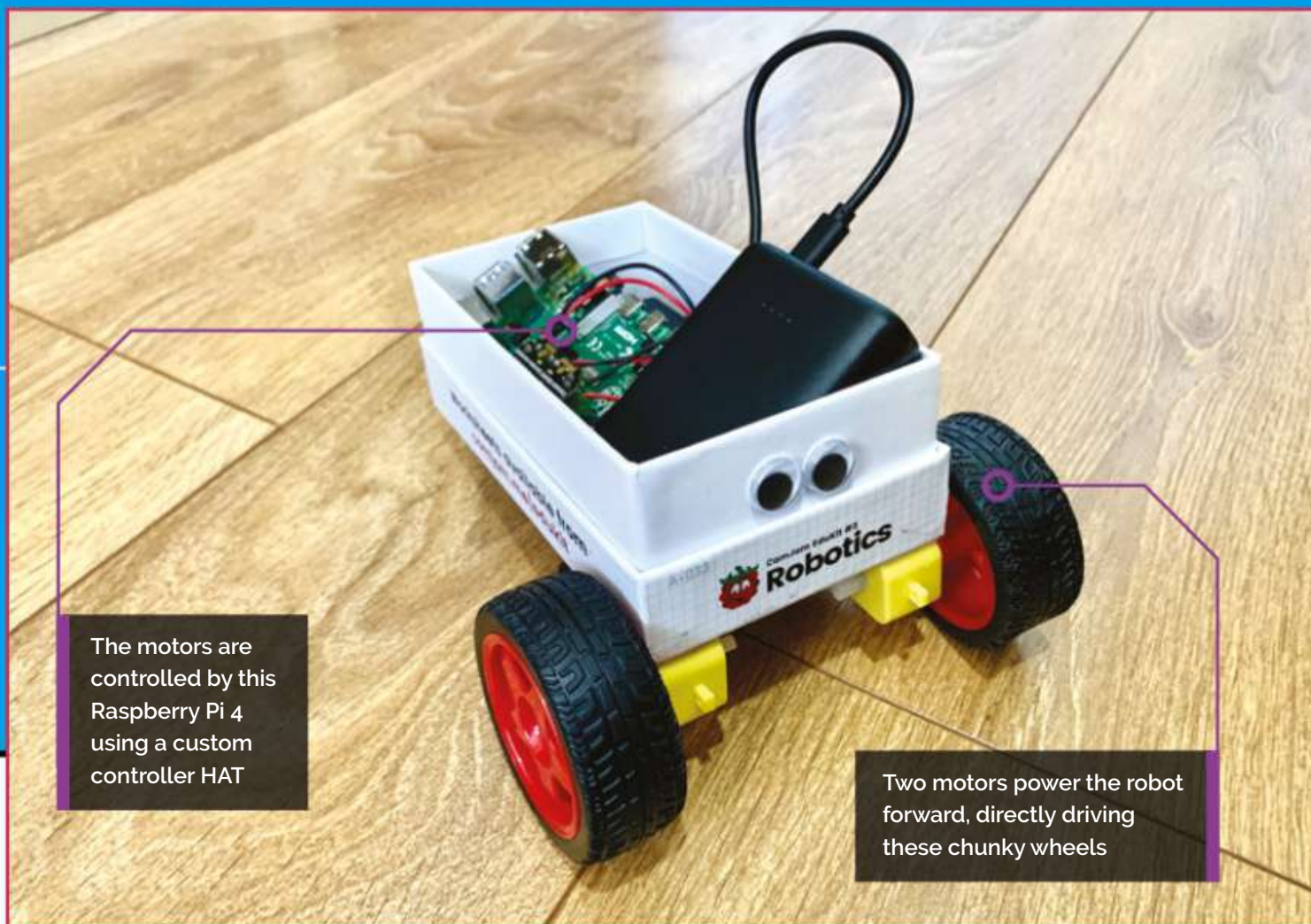
02 Prepare the brains!

Our robot, unsurprisingly, is going to be controlled by a Raspberry Pi 4. The kit works with any Raspberry Pi model, so a Zero W would also work well. We recommend using Raspberry Pi OS Lite, but everything will work with the 'regular' OS too. Using Raspberry Pi Imager (magpi.cc/imager), make sure you configure anything you need to access the OS, such as wireless LAN and SSH, using the advanced menu. Now write the image to the SD card and use it to boot your Raspberry Pi. Log in and ensure everything is up-to-date with the following command:

```
sudo apt -y update && sudo apt -y upgrade
```

03 Motor mounting

We're going to begin with the two yellow motors that will drive the wheels. Take the now-empty box, then put aside the blue cover. Flip the larger white portion over so you're looking at the base. At one of the shorter ends (doesn't matter which), attach the two motors on either



The motors are controlled by this Raspberry Pi 4 using a custom controller HAT

Two motors power the robot forward, directly driving these chunky wheels

side using the supplied sticky pads. A good tip is to trim one down to just the size of the yellow section, so we have a spare sticky for the next part. The yellow base should fit into the corner on each side, running parallel with the longer box edge, so the black part and the wires are pointing inward, and one white shaft is hanging over the edge of the box.

04 Add some steering

To help guide your little robot on its way, a castor ball has been supplied. This makes handling left and right movements much easier by being able to turn in any direction. You'll need another sticky pad (hopefully you've got a spare from the previous step). Place the sticky pad on the base of the castor ball holder, and mount it on the box base on the other side from the motors, in the middle, a little way in from the edge. Place the ball in it, if not already there. Now, carefully make a small hole next to each motor and feed the wires through. You can now flip the box over.

05 Wiring time

Motors require a lot more current than computers are normally able to safely handle. To solve this problem, CamJam has provided a special HAT add-on for your Raspberry Pi that allows the

motors to be controllable, while powered by a separate power source. Looking at the underside of the board, you'll see three green connectors. You need to connect one motor to the left connector, and the other to the right. Finally, connect the battery pack to the centre connector. Take your

“ The wrong wiring could damage your Raspberry Pi ”

time with this, as the wrong wiring could damage your Raspberry Pi. Check the diagram for the correct wiring.

06 Connect the controller

If you haven't already, shut down your Raspberry Pi 4 and disconnect it from any power. Place your Raspberry Pi 4 in the box towards the rear. Later on, you can secure it with sticky pads, but it's best to wait until you're happy with the placement and everything has been tested. Connect the controller to the GPIO, being very careful to place it on the correct pins. The controller board should point inward to the Raspberry Pi PCB and, viewing with the USB and Ethernet ports on the right, be placed on the leftmost pins (pin 1 onwards).

Top Tip

Make a chassis

Why not try and build your own chassis? Maybe you can upscale some junk and make it mobile?

► Wire up the control board as shown here. This is the view looking down

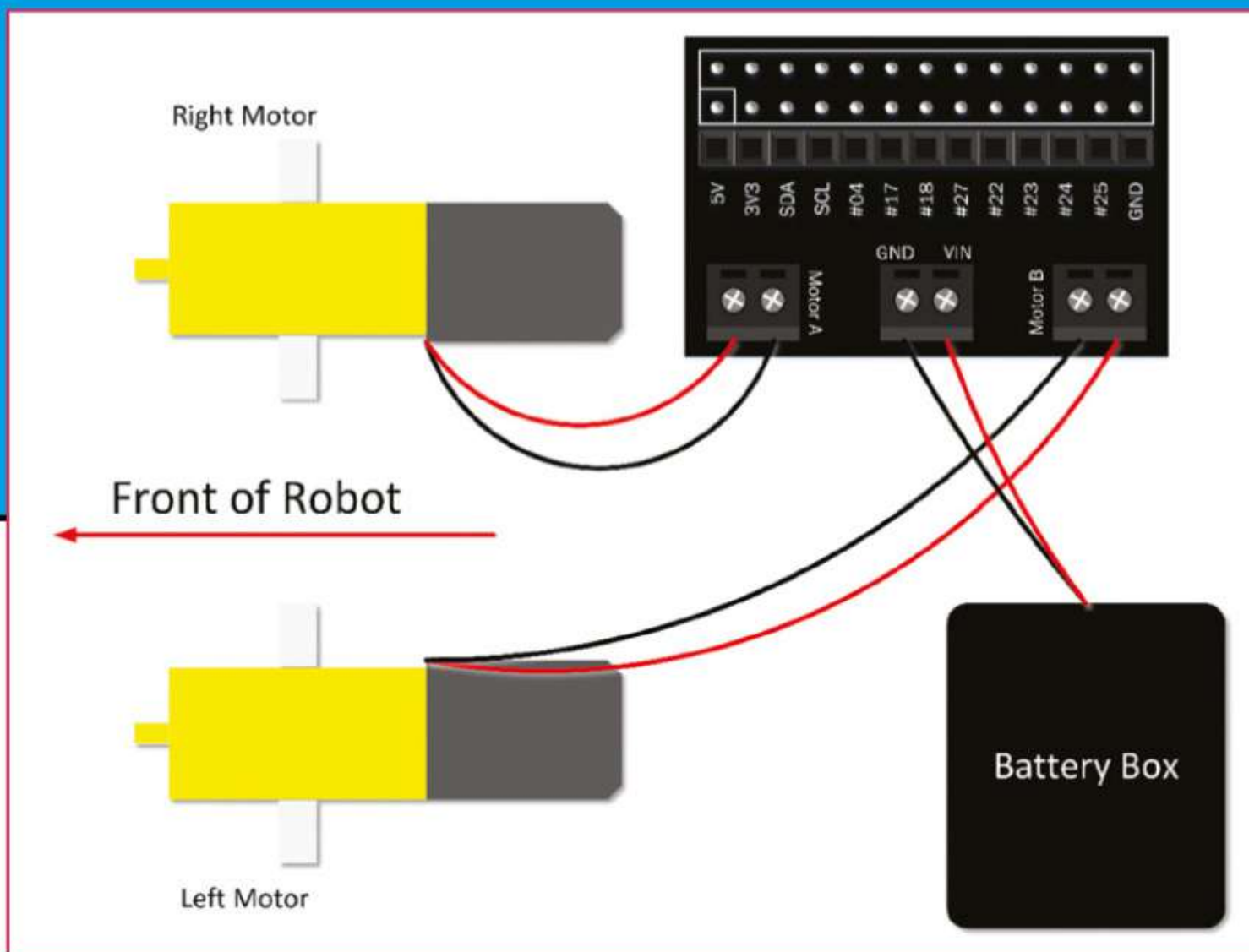
Top Tip



Zero friendly

This kit works with any Raspberry Pi model, even the original. Try a Raspberry Pi Zero W which has much lower power requirements.

▼ Make sure that the motors are positioned as shown, and that the wheels can attach without scraping the box



07 Empowering your robot

The supplied battery pack will only serve the two motors, so you need to have a think about how your Raspberry Pi 4 will be powered. If you want to power it from the mains, then the range of your new robotic pal will be limited by the length of cable you can supply. For a true free-range experience, a decent power bank, such as the type that are used to charge phones, is what you need. Make sure it's capable of supplying enough current – a capable output of 3A will be plenty. If you're using a Raspberry Pi Zero W, 1A is enough.

08 Power up

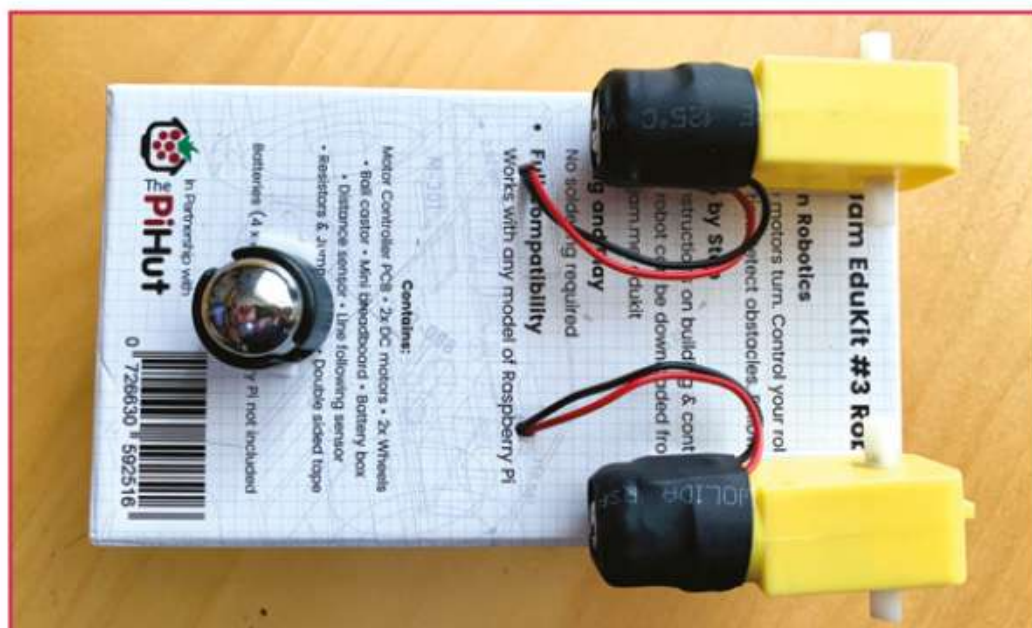
If you haven't already, insert four AA batteries into the battery pack and double-check all your wiring from the previous step. Toggle the power switch to 'on'. Absolutely nothing should happen. If anything does, such as a moving motor, switch back to 'off', and check all the wiring. Carefully place the battery box and Raspberry Pi 4 into the chassis box, placing them at the rear, to provide some stability. Now, add your power source for your Raspberry Pi 4 and power up. Don't put the wheels on just yet – it's a lot easier to fix problems if your robot cannot move!

09 Test drive

Let's make sure Raspberry Pi can control the motors. Have a look at the **test_motors.py** listing. These few lines will simply spin up the motors for a second. It uses the GPIO Zero library, which has dedicated commands for controlling this robot, and comes pre-installed with Raspberry Pi OS.

Enter the code using Thonny (or your favourite editor) and save it in your home directory as **test_motors.py**. Now run it:

```
python3 ~/test_motors.py
```



Did both motors spin? If so, you're good to carry on. Otherwise, go back through the steps and check everything again.

10 Back and forth

The next step is to check the motors are correctly wired for backwards and forwards movement. Create a new file in the home directory called **test_direction.py**, and enter the code from the **test_direction.py** listing. This will move the robot forwards, backwards, then left and right. Once you've got the code ready, run it with:

```
python3 ~/test_direction.py
```

If you can't quite see what's going on, try attaching the wheels and see your robot move for the first time! Feel free to play with the code and

“ You can control the speed ”

try different timings. Just keep an eye on any wires powering Raspberry Pi 4.

11 Make a maze!

Try out your new robot, and give yourself a coding challenge by drawing a maze for the robot to navigate. Create a new Python file, just like before, and use the forward, backward, left, and right commands to create a series of instructions that will drive the robot around the maze. You can control the speed of the robot by adding a value between 0 and 1 after each command. For example, `robot.forward(0.5)` will start the robot moving forward at half-speed. You may find this useful to stop skidding on hard floors.

12 Next month

Congratulations! You now have a small robot pal. If you'd like to find out what else we can do with the CamJam robotics kit, make sure you get next month's issue, where we will have a look at those other components we put aside. Soon, our robot will be able to detect obstacles and follow lines all on its own. Plus, we'll go beyond the kit and look at how you can customise your robot to do even more. Thanks to CamJam for providing assistance with this tutorial! 📺

test_motors.py

➤ Language: Python 3

DOWNLOAD
THE FULL CODE:



magpi.cc/testmotorspy

```
001. # CamJam EduKit 3 - Robotics
002. # Motor Test Code
003.
004. import time # Import the Time library
005. from gpiozero import CamJamKitRobot # Import the CamJam
    GPIO Zero Library
006.
007. robot = CamJamKitRobot()
008.
009. # Turn the motors on
010. robot.forward()
011.
012. # Wait for 1 seconds
013. time.sleep(1)
014.
015. # Turn the motors off
016. robot.stop()
```

test_direction.py

➤ Language: Python 3

DOWNLOAD
THE FULL CODE:

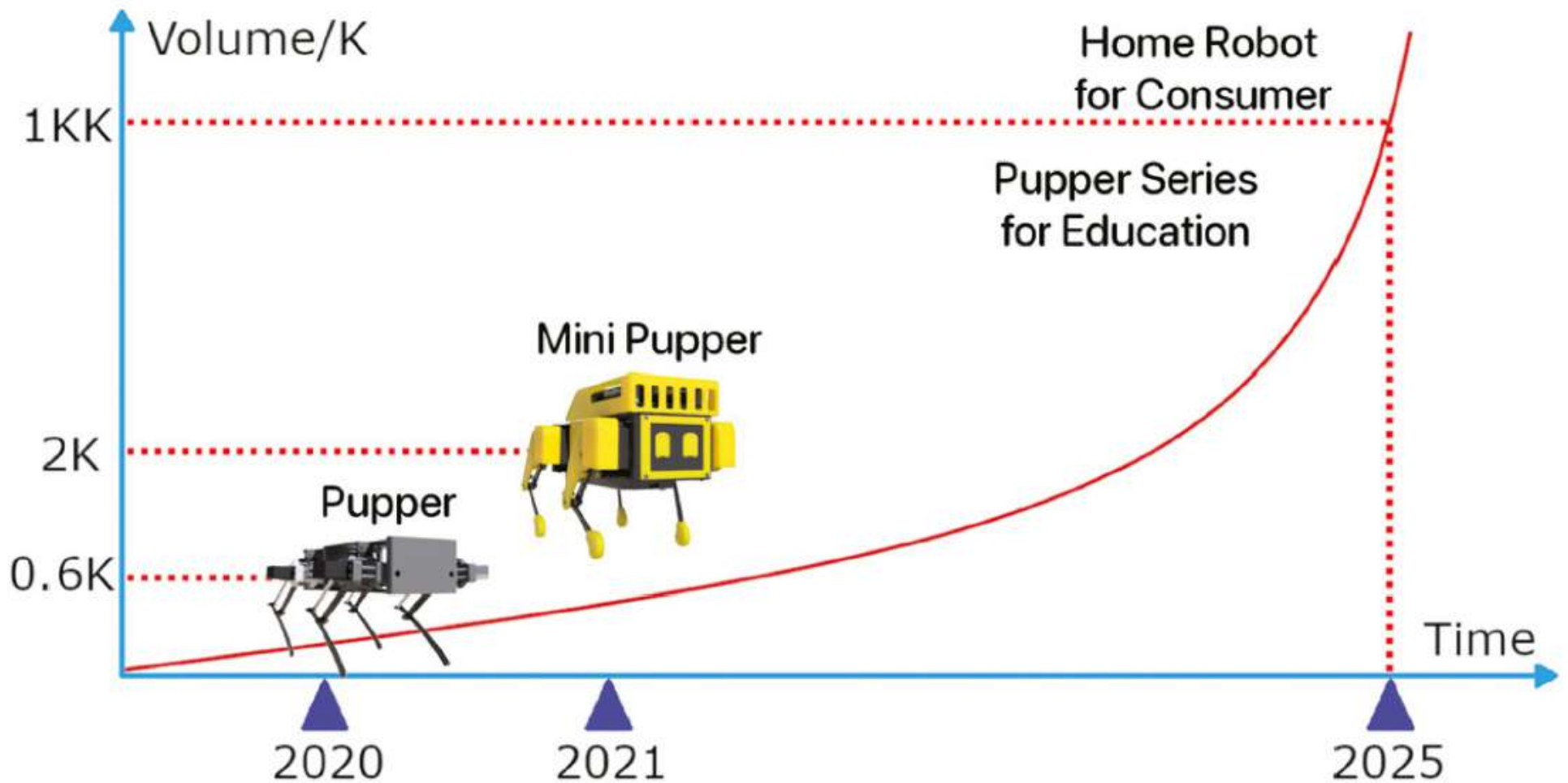


magpi.cc/testdirectionpy

```
001. # CamJam EduKit 3 - Robotics
002. # Driving and Turning
003.
004. import time # Import the Time library
005. from gpiozero import CamJamKitRobot # Import the CamJam
    GPIO Zero Library
006.
007. robot = CamJamKitRobot()
008.
009. robot.forward()
010. time.sleep(1) # Pause for 1 second
011.
012. robot.backward()
013. time.sleep(1)
014.
015. robot.left()
016. time.sleep(0.5) # Pause for half a second
017.
018. robot.right()
019. time.sleep(0.5)
020.
021. robot.stop()
```


MangDang

Everyone will own their own affordable robotics friend



Reviewed by Magazines, Websites and Customers



Workshops



Mini Pupper \$30 USD discount for Raspberry Pi community before Aug. 31, 2022

www.mangdang.net

sales@mangdang.net

MangDang

Genuine, Robot Operating System, Open-source, Education Robot Kit
Make robotics easier for schools, enthusiasts and beyond.



**Lifetime
Customer
Support**

Contact Us Anytime at
sales@mangdang.net



**3-Months
Hassle-Free
Warranty**

Straightforward for
Peace of Mind



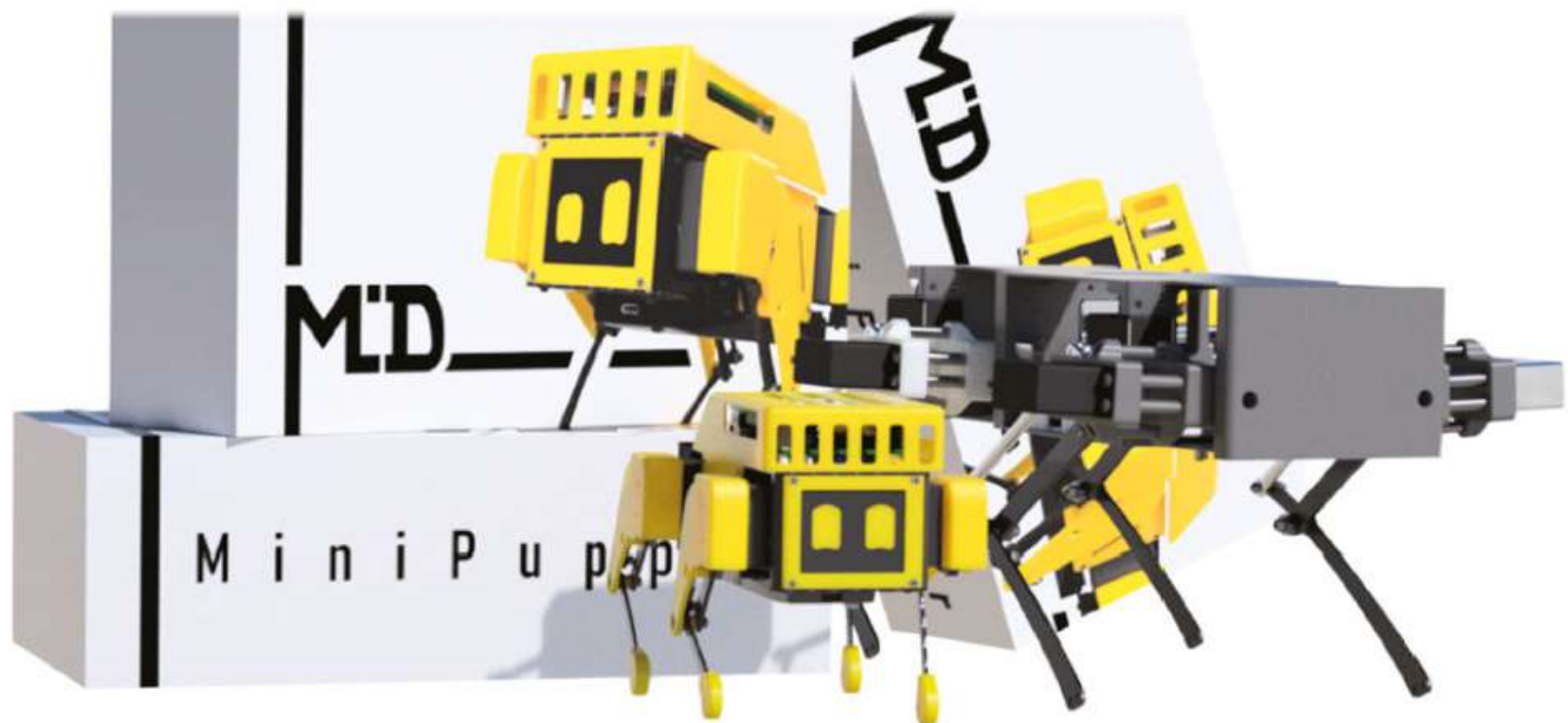
**3-Days
Global Shipping
By Air**

Ready Delivery within 3 Days
To 135+ Countries/Regions



**Online
Guides
Documents & Video**

Rich Guides Including
Documents and Videos



Easy

Enjoy DIY Journey
Assemble within 15 Mins



Official

Reviewed by
ROS, Raspberry Pi, Amazon



Media

Endorsed by
Many Global Channels



Case Study

Liked by
Schools, Enthusiasts, and Beyond

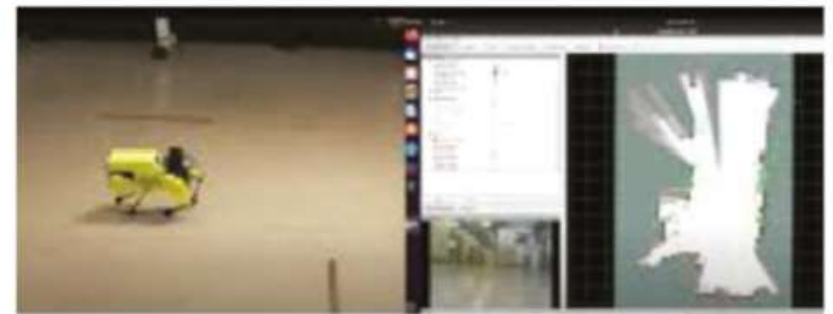
All in One



Easy Assemble



Enjoy Education



Beyond (ROS, OpenCV)

www.mangdang.net

sales@mangdang.net

Listen to the birds with BirdNET-Pi



Phil King

Long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

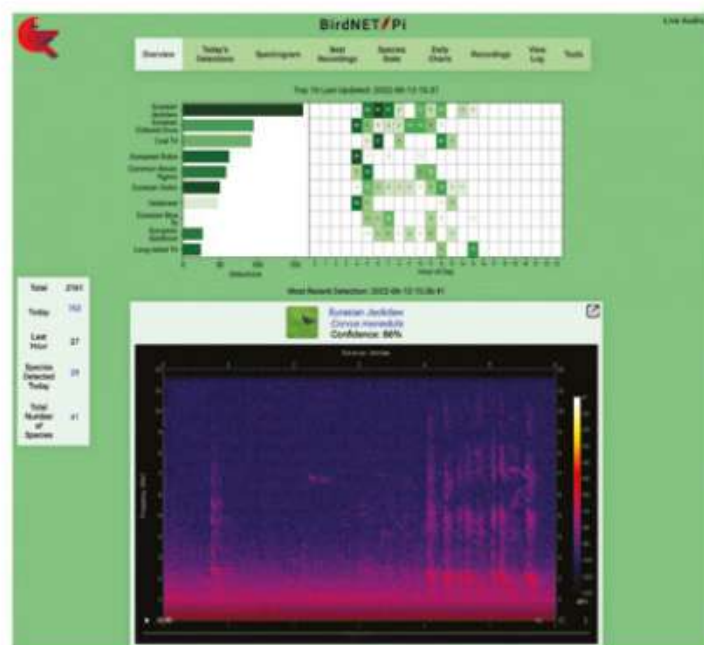
You'll Need

- ▶ Raspberry Pi 4, 3B+, or Zero 2 W
- ▶ Raspberry Pi OS 64-bit Lite
- ▶ USB microphone, or other mic with USB sound card

Identify and catalogue bird species from their calls using a Raspberry Pi and microphone

A pair of binoculars is useful for watching birds in your garden, but, as every birder knows, their calls and sounds are another key way of identifying species. That's where BirdNET-Pi comes in. Based on the BirdNET research project, Patrick McGuire's impressive piece of software runs on Raspberry Pi and is very easy to set up.

Plug in a USB mic (or other mic using a USB adapter) and it'll listen 24/7, extracting bird songs and sounds, and analysing them to identify bird species in real-time. All detections and audio recordings are catalogued automatically in a database, so you can check through them via a generated web server, which also offers a host of tools and options. You can also receive notifications and share your detections with the BirdWeather citizen science project. Let's get listening...



▶ Check out the Species Stats screen to get a breakdown of detections by time, filterable by species

01 Set up Raspberry Pi

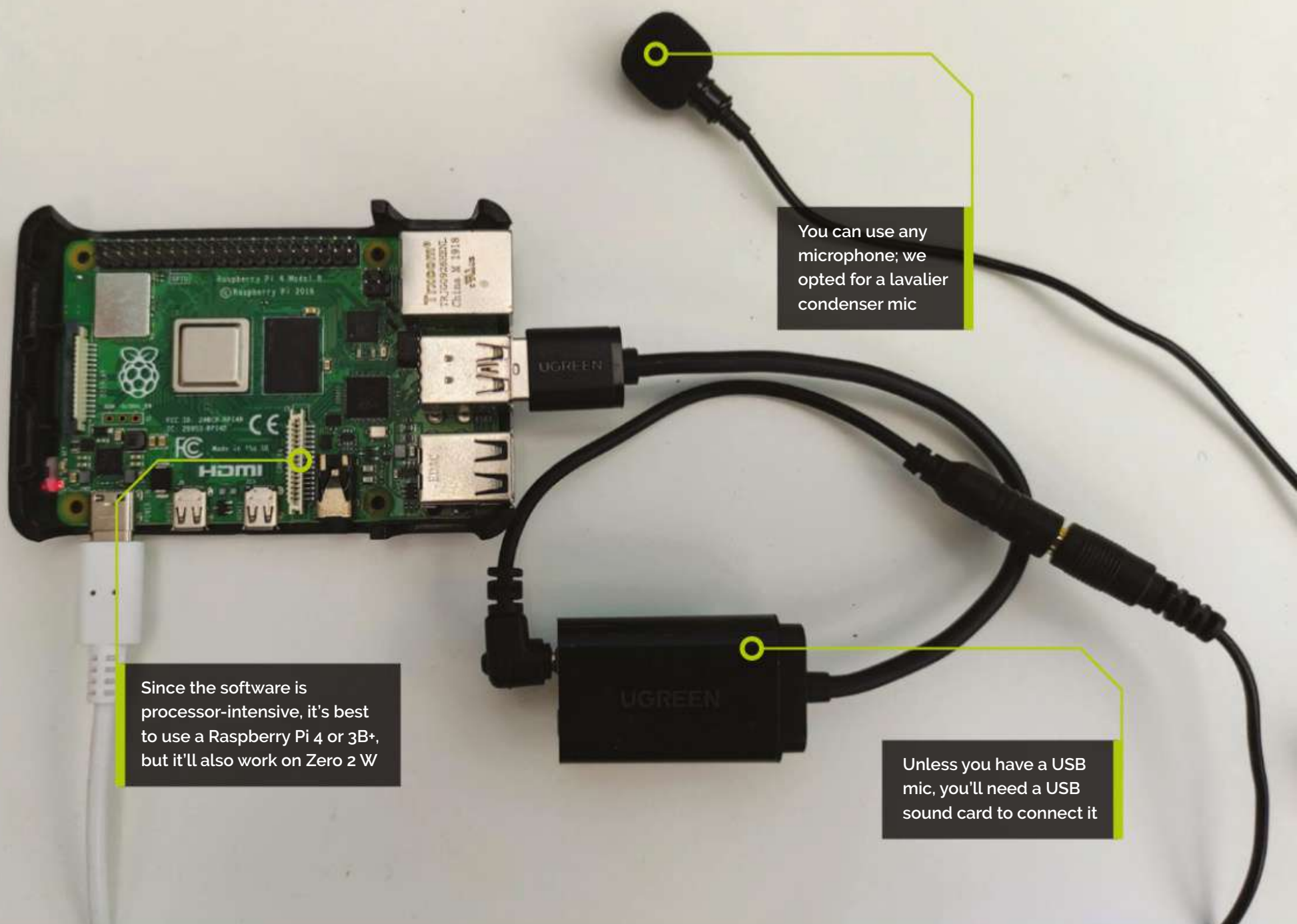
To run BirdNET-Pi, you'll need the 64-bit version of Raspberry Pi OS. As usual, you should use the Raspberry Pi Imager tool (magpi.cc/imager) on another computer to install the OS to your microSD card – 16GB or greater capacity is advised.

In Imager, click on Choose OS > Raspberry Pi OS (Other) > Raspberry Pi OS Lite (64-bit). Now click Choose Storage and select your microSD card. Click the cog icon for Advanced Options. Enable SSH, set a username and password, and configure your WiFi connection. You may also want to change the hostname to something like 'birdnetpi.local'. Finally, click Save, then Write to install the OS.

02 Install BirdNET-Pi

Booting up Raspberry Pi, you'll need to SSH in from another computer. On a Raspberry Pi 4 or 3B+, you can now install BirdNET-Pi with a single command:

▶ The Overview screen of the web interface shows the latest detections and statistics



```
curl -s https://raw.githubusercontent.com/mcguirepr89/BirdNET-Pi/main/newinstaller.sh
| bash
```

Note: If using a Raspberry Pi Zero 2 W, you'll need to enter some additional commands before

“ An omnidirectional mic is recommended as it'll capture sounds from all around ”

running that one. See magpi.cc/birdnetpiz2w for more details on Pi Zero 2 W installation.

The installation process takes 10–15 minutes. Once finished, it'll reboot Raspberry Pi and start the web server.

03 Access web server

From another computer or device, enter `http://` followed by your Raspberry Pi's hostname; e.g. <http://birdnetpi.local>. You should now see

the BirdNET-Pi web interface. First, check your location details: Click on Tools in the menu bar (the default login is **birdnet** with no password). Select Settings and check the Latitude and Longitude are correct for your location – click on the link to find your co-ordinates on a map. Click Update Settings, then return to the Overview screen.

04 Plug in a mic

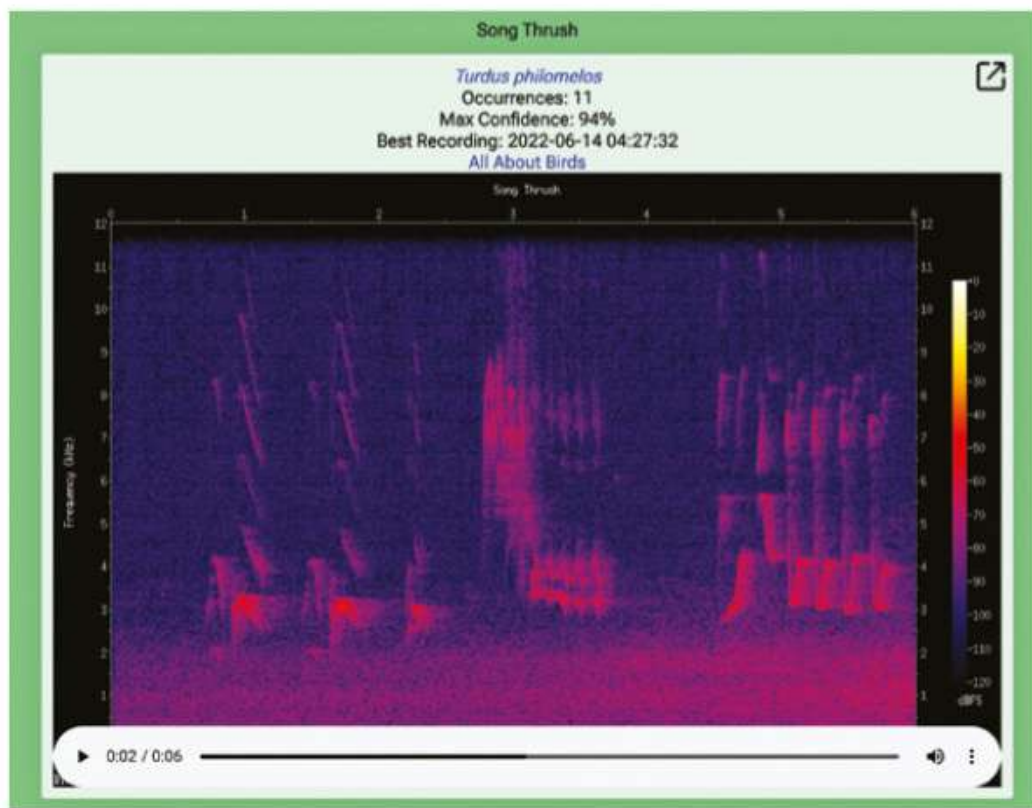
An omnidirectional mic is recommended as it'll capture sounds from all around. We started off with a low-cost mini mic USB dongle plugged into a USB extender cable. While this worked, it wasn't really sensitive enough and also had a lot of background noise.

We upgraded to a lavalier condenser mic combined with a USB sound card. You could use a similar setup or splash out a little more on a better mic or even build a DIY one (see magpi.cc/birdnetpimics). If using a USB sound card, make sure its mic port wiring type matches that of your mic's 3.5 mm jack – we needed to use a TRRS to TRS adapter for ours.

Top Tip

Get notified

BirdNET-Pi supports notifications for over 70 services, including email and Twitter. Click on Apprise Notifications on the Settings page for more details on the configuration needed for each service.



▲ A spectrogram for each bird detection can be viewed, and its audio recording extract played

05 Testing, testing

To check the mic is working correctly, click on Live Audio (top right) in the web interface, then the play arrow icon, and you should hear whatever it's picking up.

” The software will run 24/7 while Raspberry Pi is turned on, listening for bird calls ”

Before placing the mic outside, you may want to test the system by playing a few bird sounds to it from YouTube or elsewhere. The software should be able to detect the correct species, in which case a new detection will soon show up on the Overview screen, along with a spectrogram and audio recording.

Once you're happy it's all working well, it's time to move your mic outside. You could place it on a stand, but we just hung ours from an open window. Unless using a weather-proof case, Raspberry Pi should stay indoors.

06 Adjust sound level

You can adjust the USB input level for the mic if needed. Use the Tools > Web Terminal option in the web interface, then log in with your username and password to SSH in. Enter the command `alsamixer -m` to access the ALSaMixer sound settings.

Click 'Select sound card' (F6), then USB Audio Device. Click Capture (F4) and adjust the mic's input level. You want it high enough to be able

▶ Link your listening station up to the BirdWeather citizen science project to view it on the map there

to hear the bird calls, but avoiding excessive background noise.

Tip: if you have the Spectrogram displayed on another device, such as a phone, you can see/hear the effect of the adjustments.

07 View detections and info

The software will run 24/7 while Raspberry Pi is turned on, listening for bird calls. The most recent five detections appear on the Overview screen. At the top, you'll also see a chart for the top ten species detected, with totals for each hour of the day. The Daily Charts menu option lets you view charts for different dates.

Click on Today's Detections to scroll through them all, or search by species. The Species Stats option shows stats in charts, filterable by species. Click the Recordings option and select By Species or By Date to browse recordings. To see what's going on under the hood, click the View Log option to see BirdNET-Pi analysing the latest audio recording.

08 Add images

You can add images automatically for birds from the Flickr service. To get the required Flickr API key, go to Tools > Settings and click on 'Get your free key here'; on the Flickr App Garden page, click 'Apply for your key online now', then 'Apply for a non-commercial key'. Fill in the details and click Submit to get your key (and secret, not needed here). Paste the key into the Flickr API Key field on the BirdNET-Pi Settings screen, then click Update Settings.

A photo will now appear for every detection. Clicking on the bird's English name on the Overview will show you its latest spectrogram



and recording along with a gallery of photos. Clicking on the bird's Latin name takes you to its Wikipedia page so you can find out all about it.

09 Link to BirdWeather

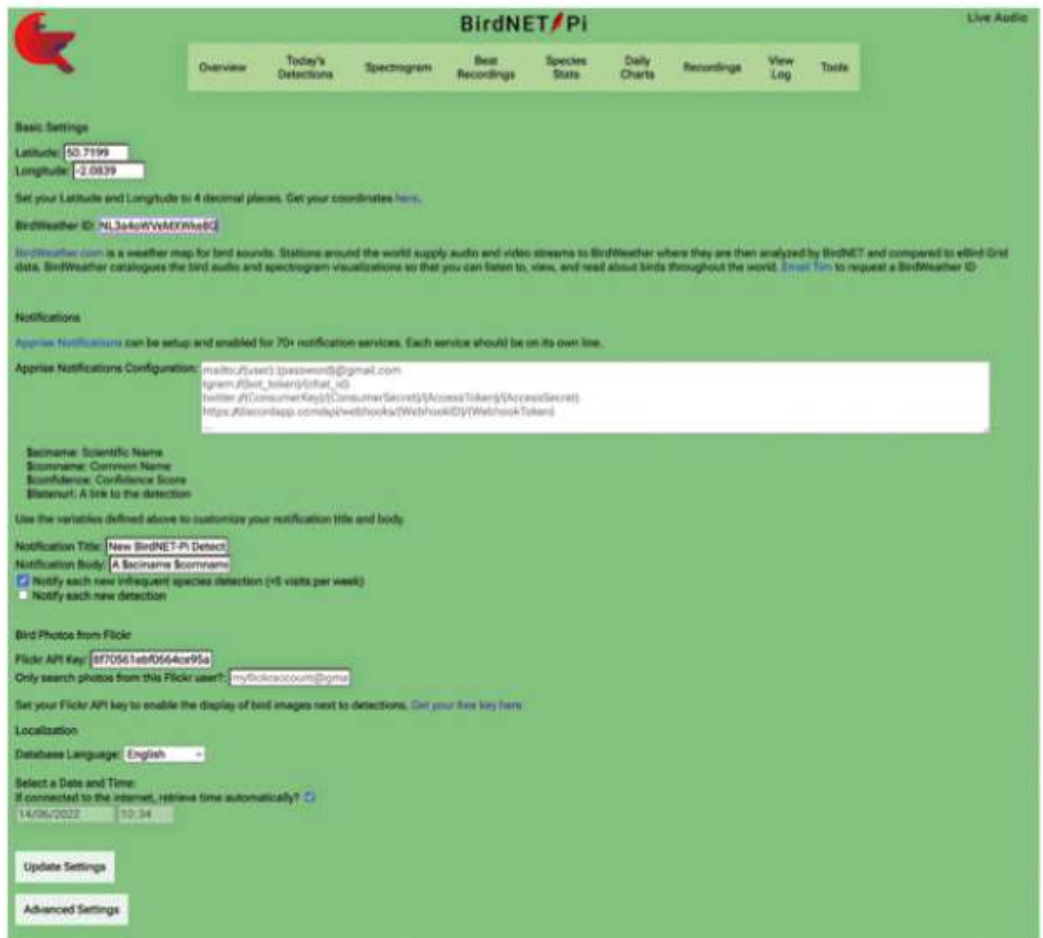
You can link your BirdNET-Pi station up to the BirdWeather citizen science project. In Tools > Settings, click 'Email Tim' to send an automatically generated email to him and you should receive a response within 24 hours with your ID – paste it into the BirdWeather ID field, then click Update Settings. Your station will now appear on the BirdWeather site (app.birdweather.com), showing its detections in real-time.

10 Advanced settings

By default, the minimum confidence level for a detection is 0.7 (70%), but this can be altered. Go to Tools > Settings and click Advanced Settings, then adjust the Minimum Confidence Level. Other settings on this screen include recording/extraction length, sigmoid sensitivity, and the Privacy Threshold to prevent human voices being logged and recorded.


11 Manage misidentifications

Occasionally, BirdNET-Pi may misidentify a bird. If this occurs, you can access the database to remove the erroneous entry. Go to Tools > Database



Maintenance and click Login. Then search for the species, click on 'detections', and delete one or more entries.

If a bird keeps being detected erroneously, you can tell BirdNET-Pi to ignore it. Go to Tools > Excluded Species List and add it to the list. To erase audio recordings for a species, go to Tools > Web Terminal and enter the remove command followed by its name, e.g. `remove Atlantic Puffin`. You can use `*` as a wild card, but be careful not to erase birds you want to keep!

Don't be too quick to dismiss unlikely detections, either: you may well find that you have several bird species in your locale that you've never spotted – but you can hear them! 

▲ The Basic Settings screen enables you to get photos from Flickr, link up to BirdWeather, set up notifications, and more

Top Tip

Storage space

With a decent mic equipped, BirdNET-Pi will typically make hundreds of recordings daily. Eventually it'll fill your microSD card, at which point older recordings are automatically deleted – unless you protect selected ones by clicking the padlock icon next to them.

◀ The View Log screen shows what's going on under the hood in real-time, as recordings are analysed and detections made



Build a 64-bit Minecraft server



MAKER PJ Evans

PJ is a writer, tinkerer, and software engineer. His son made him write this tutorial and has been online ever since.

twitter.com/mrpjevans

The latest Minecraft update brings great new features, but also new system requirements. Use 64-bit Raspberry Pi OS to set up a multiplayer server

Minecraft has been a feature of the Raspberry Pi ecosystem since the early days. Although the game is demanding on resources, a cut-down educational version became available and was very popular. As Minecraft has developed with new features, challenges, and mods, the 32-bit world was left behind – especially as the most recent release was 64-bit only, owing to the version of Java required. With the release of Raspberry Pi OS 64-bit, the little computer that can is back in the game with all the power of the latest Raspberry Pi 4 too. We’re going to show you how to build a Minecraft multiplayer server for you and your friends to enjoy.

01 Set up your hardware

Setting up a Minecraft server may not seem like a hardware project, but the performance of the server is critical to fun gameplay. Firstly, we need the best Raspberry Pi can offer, and that’s a Raspberry Pi 4. You can use an earlier model if you wish, but you won’t get the smooth gameplay that the latest and greatest can offer you. A big factor is memory. Java uses a lot of memory and Minecraft needs still more. If you can, get an 8GB model, 4GB minimum. Finally, avoid wireless LAN if at all possible. Use a hardwired connection to your router if you can.

02 Prepare your SD card

We’ve suggested using a USB-boot M.2 SSD drive for best performance, but you can use a regular SD card too. Just make sure it’s high quality as it will be heavily used by Minecraft. Either way, the process is the same: using Raspberry Pi Imager

Network lag is the enemy of good gameplay, so avoid wireless LAN and use a wired connection to your router



The Minecraft server puts a lot of strain on a Raspberry Pi 4B, so a good case with built-in cooling is essential

(magpi.cc/imager) select Choose OS > Raspberry Pi OS (other) > Raspberry Pi OS Lite (64-bit). Now select your card under ‘Storage’, then click the cog icon. Set the hostname (we used ‘steve’), enable SSH, set a password for the user ‘pi’, and configure your wireless LAN (if using). Now write your image.

03 Install dependencies

A dependency is a piece of software that is needed by the software we want to run. In our case, this is Java, a popular programming language

You’ll Need

- Raspberry Pi 4 case with cooling
magpi.cc/casefan
- Minecraft Java Edition
minecraft.net
- Spare router port (optional)
- Ethernet cable (optional)
- M.2 SSD drive (optional)

and platform that was used to write Minecraft. The latest version of the Minecraft server requires version 17 or above, which is only available as 64-bit. Before installing, log into the OS and make sure everything is up-to-date with:

```
sudo apt -y update && sudo apt -y upgrade
```

Once complete, install the Java runtime:

```
sudo apt install openjdk-17-jdk git
```

(We're also installing Git, as we'll need it later). Test everything has worked:

```
java --version
```

If you get the version number back, you're good to go.

“ The performance of the server is critical to fun gameplay ”

04 Create a dedicated user

As we may want others to join our Minecraft games, it makes sense, and good practice, to run Minecraft under a dedicated user account with restricted permissions. That way, if someone gains access to the server via Minecraft, they will be heavily restricted on what they can do. To create the user:

```
sudo useradd -r -m -U -d /usr/local/minecraft -s /bin/bash minecraft
```

We now need to assume that user's identity to continue. To do this we can 'swap user' or 'su':

```
sudo su - minecraft
```

Now, we'll create some local directories to store the things we need:

```
mkdir -p ~/backups,tools,server
```

This neat trick allows us to create several directories with one command.



05 Build the console tool

Minecraft comes with a control interface called rcon. So that Minecraft can be shut down tidily when we shut down the computer, we need to ask Minecraft to save and stop, using rcon. A neat utility called mcrcon helps us do this. We need to download the source code and build it, so let's start with downloading the code from GitHub. As the 'minecraft' user:

```
cd ~/tools
git clone https://github.com/Tiiffi/mcrcon.git
```

Now we can build the utility:

```
cd ~/tools/mcrcon
gcc -std=gnu11 -pedantic -Wall -Wextra -O2 -s -o mcrcon mcrcon.c
```

To check it works, run the help screen:

```
./mcrcon -h
```

06 Download the Minecraft server

We now have everything we need to install and run Minecraft. It's time to download the server and set it up. To find the latest version, visit magpi.cc/mcserver, right-click on the download link, and 'Copy link'. Using that link, and as the 'minecraft' user, download the server directly into the ~/server directory on Raspberry Pi 4:

▲ Start Minecraft Java Edition and then select Multiplayer from the main menu

Top Tip

Get an M.2 boost

For a speed boost, replace the SD card with an M.2 SSD. Cases such as the Argon M.2 support them as a USB boot device.



▲ You're online in a virtual world with your friends

Top Tip



VPN

An alternative to having an open Minecraft server port is to install a VPN server such as WireGuard (piVPN) and allow friends to connect to it.

► If your client can't find the server, don't worry, just select 'Direct Connection'

```
cd ~/server
wget https://launcher.mojang.com/v1/objects/
e00c4052dac1d59a1188b2aa9d5a87113aaf1122/
server.jar
```

Note: this link is the latest - v1.19 - at the time of writing.

Now, run the server for the first time. It will configure a lot of things and then exit:

```
java -Xmx1024M -Xms512M -jar server.jar
nogui
```

07 Configure the server

Before you can go around chasing creepers, we need to set a few things up on the server. You may have spotted a message about the end-user licence agreement (EULA) on the previous run. We need to set that (to show we agree) and change a couple of things to enable the rcon protocol.



```
nano ~/server/eula.txt
```

Change false to true, then save and exit (**CTRL+X**).

```
nano ~/server/server.properties
```

Find `rcon.password=` and add a strong password of your choosing (remember it!), then find `enable-rcon=` and change `false` to `true`. Now save and exit.

08 Test it!

Let's try it out. The command to start Minecraft manually is:

```
java -Xmx4096M -Xms512M -jar server.jar
nogui
```

The number after `-Xmx` is the maximum amount of memory Minecraft can use. Here, it is 4GB (4096MB). Start with this and then try increasing it by 1GB at a time if you run into trouble.

Run up the server using the command above and, once running, see if you can connect to it using Minecraft Java Edition on your device of choice. Click on 'Multiplayer' then 'Direct Connection' then enter the server name, in our case 'steve.local'. After a few seconds, you should be dropped into the world and you're online! **CTRL+C** stops the server.

09 Run as a service

Ideally, we'd like the Minecraft server to run on boot, so we don't have to log in and start it manually. We can create a system service to do this. Start by creating the configuration file:

```
exit # If you're still the minecraft user
sudo nano /etc/systemd/system/minecraft.
service
```

Add the contents of the **minecraft.service** listing on this page (or copy from **magpi.cc/minecraftservice**). Remember to change `password` to the actual password you thought of in the previous step. Now enable the service:

```
sudo systemctl enable minecraft
```

Finally, `sudo reboot` to restart the OS and you



▲ Enter the hostname of your server followed by '.local'. If joining from outside the network, this needs to be your router IP address

can check that Minecraft is running by checking its status:

```
sudo systemctl status minecraft
```

“ Investigate services such as DynDNS ”

10 Configure your router

Your Minecraft server is now up and running, but no one outside of your home can access it! To allow friends and family to play along, we need to ask your home router to permit access. This is something to be done cautiously, as you do not want to accidentally allow people access to parts of your network. Consult the instructions for your router, and follow these steps: first, set up a DHCP reservation for your Minecraft server; this means it will never change IP address. Secondly, set up port forwarding from port 25565 on the router to the same port on your server. You may need to restart your router for the changes to take effect.

11 Testing access

For others outside your network to play, you'll need to give them your public IP address (your router will tell you this). It may take a couple of goes to get access sorted out. One issue you may come across is your router having a dynamic IP address, so occasionally the IP address it presents

to the world changes to something new. If this becomes a pain, investigate services such as DynDNS that assign a domain to your IP which updates automatically if your IP changes. Now, an external user should be able to join your server using your public IP address.

12 Securing access

You now have a publicly available Minecraft server. That might not be a good idea! To stop anyone, including strangers, joining, we recommend implementing the whitelist feature. This will allow only named players (i.e. people you know) from joining. In Minecraft itself, you can enable the whitelist with `/whitelist on` and then control access using `/whitelist add <user>` and `/whitelist remove <user>`. This is an essential step to keep your server safe. Also, consider changing the number of your external port to something else, as this can help dodge bots scanning for Minecraft servers. Have fun!

Thanks to Heidy Ramirez of shells.com for their excellent blog post on this subject. 📖

minecraft.service

► Language: **Shell**

**DOWNLOAD
THE FULL CODE:**



magpi.cc/minecraftservice

```
001. [Unit]
002. Description=Minecraft Server
003. After=network.target
004.
005. [Service]
006. User=mincraft
007. Nice=1
008. KillMode=none
009. SuccessExitStatus=0 1
010. ProtectHome=true
011. ProtectSystem=full
012. PrivateDevices=true
013. NoNewPrivileges=true
014. WorkingDirectory=/usr/local/minecraft/server
015. ExecStart=/usr/bin/java -Xmx4096M -Xms512M -jar
    server.jar nogui
016. ExecStop=/usr/local/minecraft/tools/mcrcon/mcrcon -H
    127.0.0.1 -P 25575 -p password stop
017.
018. [Install]
019. WantedBy=multi-user.target
```


Learn ARM assembly: access Pico temperature sensor



Stephen Smith

Stephen is a retired software developer who has written three books on ARM assembly language programming. He is a member of Sunshine Coast Search and Rescue and enjoys mountain biking, hiking, and running. He is also a member of the Sunshine Coast Writers and Editors Society (scwes.ca).

magpi.cc/stephensmith

You'll Need

- Raspberry Pi
- Raspberry Pi OS 32-bit
- Raspberry Pi Pico
- Raspberry Pi Pico SDK

Learn how to access hardware directly from assembly language on Raspberry Pi Pico

Raspberry Pi Pico contains a wealth of integrated hardware that is normally accessed via the Pico's SDK, but what is the magic software that controls the hardware?

We don't need to learn any new instructions to access the hardware, because all the access is via special memory addresses. These memory addresses are connected to the hardware, rather than to memory, and writing to these addresses either configures the hardware or transfers data to the hardware. Similarly, if you read these addresses, you retrieve the hardware status or read data from the hardware.

RP2040 is the chip at the heart of Raspberry Pi Pico that contains a built-in temperature sensor, which is a simple analogue device that measures the voltage drop across a biased bipolar diode, with its resistance changes proportional to temperature. It is connected to Pico's analogue-to-digital converter (ADC). The ADC reads the voltage and converts it to a 12-bit digital value. In this tutorial, we access Raspberry Pi Pico's built-in temperature sensor that is directly connected to the ADC's port 4. In this case, we use two hardware registers accessed by memory locations:

0x4004c000 – the ADC control and status register

0x4004c004 – the ADC result register

We then convert this raw 12-bit number into the temperature in degrees Celsius, using the formula:

$$\text{Temp } ^\circ\text{C} = 437 - (100 \times \text{rawADCvalue}) / 215$$

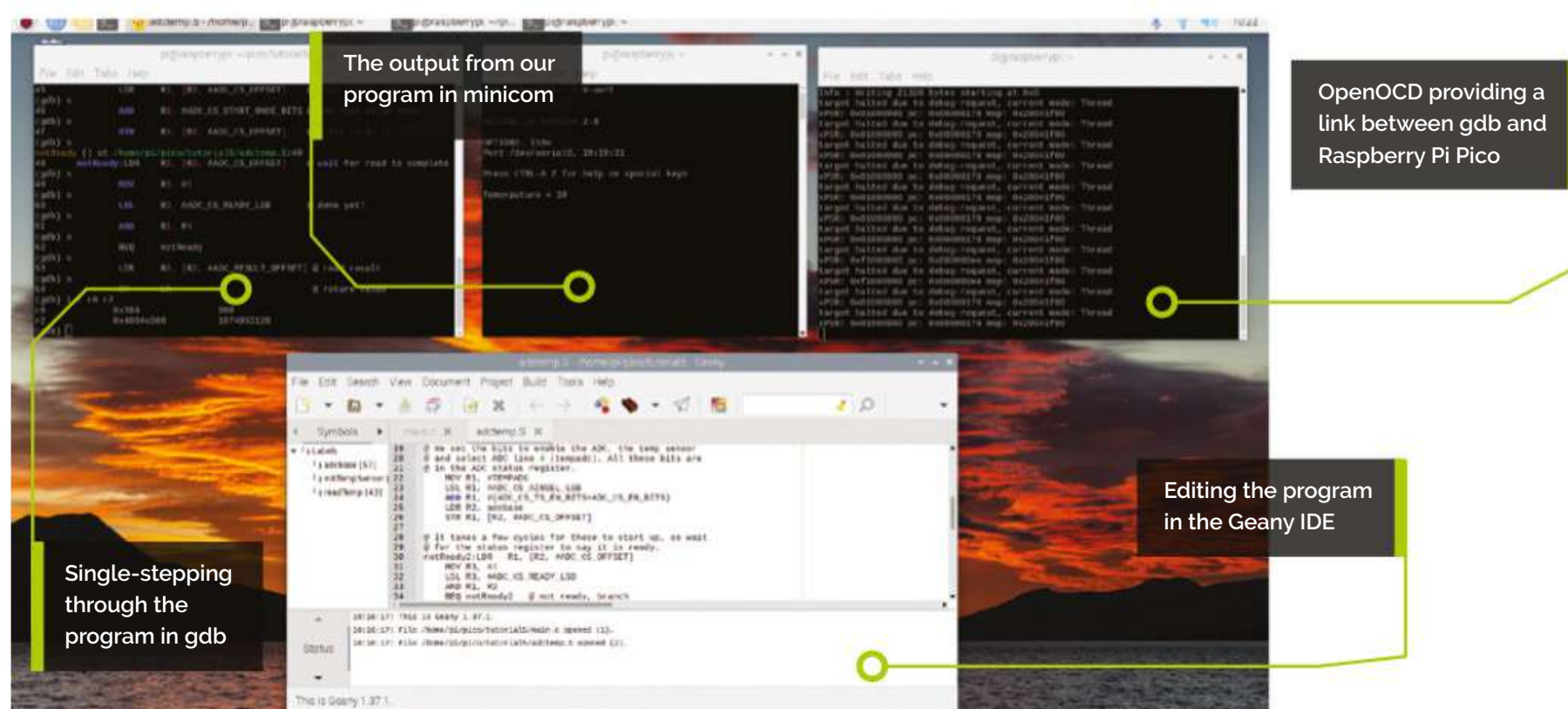
The program performs this computation in $^\circ\text{C}$ to avoid talking about the built-in division coprocessor. This is a version of the formula that avoids using floating-point arithmetic.

Full details of Pico's temperature sensor are contained in the RP2040 datasheet (magpi.cc/rp2040datasheet), section 4.9.5.

If you use a device connected to a GPIO pin on Pico, you need to configure the advanced peripheral bus (APB) to connect the device to a pin, but this isn't needed with the internal temperature sensor.

01 Create the program

Create a folder named **tutorial5** in the **pico** folder that was created in your home folder by the Raspberry Pi Pico SDK's setup script. The source code for this tutorial is in **main.c**,



adctemp.S, and **CMakeLists.txt**. Copy the file **pico_sdk_import.cmake** from the SDK's external folder to the **tutorial5** folder. In this folder, create a new folder called **build**. The **tutorial5** folder should now look like:

```
pi@raspberrypi:~/pico/tutorial5 $ ls -l
total 20
-rw-r--r-- 1 pi pi 1634 May 25 13:25
adctemp.S
drwxr-xr-x 6 pi pi 4096 May 25 13:25 build
-rw-r--r-- 1 pi pi 422 May 25 13:10
CMakeLists.txt
-rw-r--r-- 1 pi pi 476 May 25 13:23 main.c
-rw-r--r-- 1 pi pi 2763 Apr 9 10:37 pico_
sdk_import.cmake
```

The capital S in the file extension to **adctemp** is important because it allows the inclusion of C header files from the Pico's SDK.

02 Build the program

These steps are identical to those in part 3 of this tutorial series (magpi.cc/118). The difference is that the **CMakeLists.txt** file adds a C source file. Open a Terminal window and change directory to the **tutorial5** folder.

```
cd pico/tutorial5/build
```

Run **cmake** with the option to perform a debug build, to step through the program in the **gdb** debugger in a later step. Placing this command in a script file in the **\$HOME/bin** folder with a short filename can be a real time-saver.

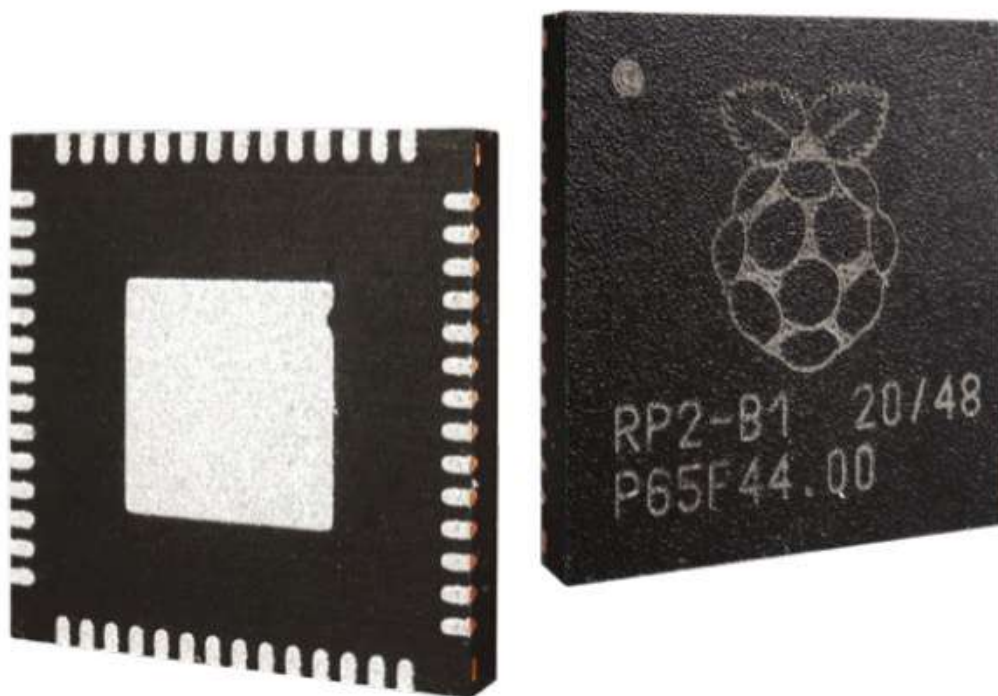
```
cmake -DCMAKE_BUILD_TYPE=Debug ..
```

The **cmake** command doesn't build the program – instead it creates a makefile used to compile by running the **make** command:

```
make
```

The **build** folder should now look like the example below:

```
pi@raspberrypi:~/pico/tutorial5/build $ ls
-l
total 1120
-rw-r--r-- 1 pi pi 18812 May 25 13:11
CMakeCache.txt
drwxr-xr-x 5 pi pi 4096 May 25 13:25
CMakeFiles
-rw-r--r-- 1 pi pi 1670 May 25 13:11
cmake_install.cmake
drwxr-xr-x 6 pi pi 4096 May 25 13:12
elf2uf2
drwxr-xr-x 3 pi pi 4096 May 25 13:11
generated
-rw-r--r-- 1 pi pi 75136 May 25 13:11
Makefile
drwxr-xr-x 6 pi pi 4096 May 25 13:11
pico-sdk
-rwxr-xr-x 1 pi pi 21320 May 25 13:25
temperature.bin
-rw-r--r-- 1 pi pi 343148 May 25 13:25
temperature.dis
-rwxr-xr-x 1 pi pi 316820 May 25 13:25
temperature.elf
-rw-r--r-- 1 pi pi 229840 May 25 13:25
temperature.elf.map
-rw-r--r-- 1 pi pi 60033 May 25 13:25
```

▲ Raspberry Pi's RP2040 chip contains a temperature sensor that we can access directly in assembly language

```
temperature.hex
-rw-r--r-- 1 pi pi 43008 May 25 13:25
temperature.uf2
```

03 Run the program

Power-on Raspberry Pi Pico by plugging the USB cable into Raspberry Pi, while holding the BOOTSEL button. When the file explorer window appears, open it and copy the file **temperature.uf2** to Pico. Pico reboots, then the program runs. The program prints out 'Temperature = 20'. Open a serial port program to see this output.

```
minicom -b 115200 -o -D /dev/serial0
```

The **minicom** command displays data being sent from the Pico to Raspberry Pi.

04 Debug the program

Part 3 of this tutorial series, in *The MagPi* #118 (magpi.cc/118), contained more details on the debugging process; this article contains a skeleton of the required operations. First, configure gdb to talk to OpenOCD, by creating a file **.gdbinit** in the home folder. Once **.gdbinit** is created, place the following line in it:

```
target remote localhost:3333
```

Next, open a Terminal window to run OpenOCD. Enter the following command in the Terminal window:

```
openocd -f interface/raspberrypi-swd.cfg -f
target/rp2040.cfg
```

We can debug our program with the following command:

```
gdb-multiarch temperature.elf
```

To load the program, enter the **load** command:

```
(gdb) load
```

05 Step through the program

The program executes Raspberry Pi Pico SDK's startup code. To skip this part and step through the program in our code, we will set a breakpoint at our main routine.

```
(gdb) b main
Breakpoint 1 at 0x1000035c: file /home/pi/
pico/tutorial5/main.c, line 14.
Note: automatically using hardware
breakpoints for read-only addresses.
```

Now, when we execute the program using the **continue** command, execution will initialise the RP2040 and Pico's SDK before halting at the start of the main routine.

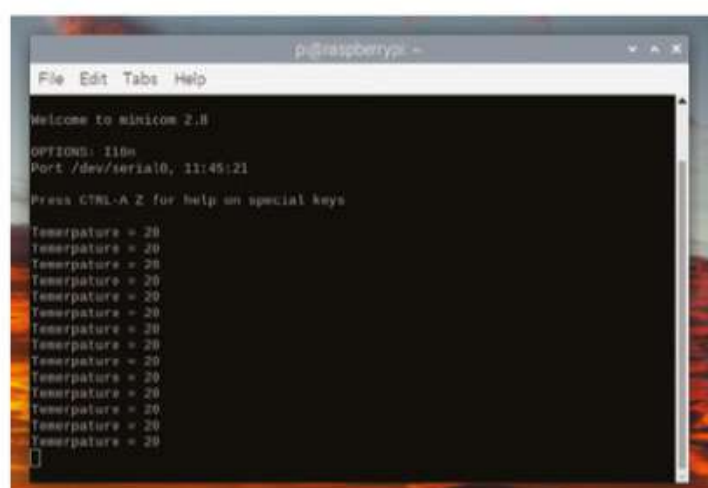
```
(gdb) c
Continuing.
target halted due to debug-request, current
mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp:
0x20041f00
```

Execution will initialise the RP2040 and Pico's SDK

```
Thread 1 hit Breakpoint 1, main () at /home/
pi/pico/tutorial5/main.c:14
14 {
```

Execute a **step** command to move to the **stdio_init_all()** function call. Use **step** again into this routine, then enter a **next** command to continue to the **initTempSensor()** routine.

```
(gdb) s
20 stdio_init_all();
(gdb) s
target halted due to debug-request, current
mode: Thread
```

▲ Running our program and observing the output in minicom

```
xPSR: 0x01000000 pc: 0x00000178 msp:
0x20041f00
  stdio_init_all ()
    at /home/pi/pico/pico-sdk/src/rp2_
common/pico_stdio/stdio.c:269
269   stdio_uart_init();
(gdb) next
target halted due to debug-request, current
mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp:
0x20041f00
main () at /home/pi/pico/tutorial5/main.c:22
22  initTempSensor();
```

06 Configure the ADC

Stepping into the `initTempSensor()` routine switches to assembly language. This routine enables the ADC and configures it to report on port 4 – the temperature sensor. The code in `adctemp.S` starts with two `include` directives:

```
#include "hardware/regs/addressmap.h"
#include "hardware/regs/adc.h"
```

These are C include files; however, using the capital S on the file extension tells GCC to run the C preprocessor on our source file. If these header files only include C processor commands like `#define` ones, then this is fine. If they include C source code, then it will result in errors. Many of the C header files included in Raspberry Pico's SDK are written to be used from either C or assembly language. In this case, we want the base memory address for the ADC's registers from `addressmap.h`:

```
#define ADC_BASE _u(0x4004c000)
```

Then we want information on the offsets of the registers from the base, as well as information on the various bits within these registers from `adc.h`. These are the values used in the program:

CMakeLists.txt

► Language: CMake

```
001. //
002. // C program to call our Assembly Language
003. // to upper routine.
004. //
005.
006. #include <stdio.h>
007. #include "pico/stdlib.h"
008.
009. extern void initTempSensor();
010. extern int readTemp();
011.
012. #define MAX_BUFSIZE 255
013. void main()
014. {
015.     char *str = "This is a test.";
016.     char outBuf[MAX_BUFSIZE];
017.     int adctemp;
018.     int temp;
019.
020.     stdio_init_all();
021.
022.     initTempSensor();
023.
024.     while( 1 )
025.     {
026.         adctemp = readTemp();
027.         temp = 437 - (100 *adctemp) / 215;
028.         printf("Temperature = %d\n", temp);
029.         sleep_ms(1000);
030.     }
031. }
```

```
#define ADC_CS_OFFSET _u(0x00000000)
#define ADC_CS_AINSEL_LSB _u(12)
#define ADC_CS_EN_BITS _u(0x00000001)
#define ADC_CS_TS_EN_BITS _u(0x00000002)
#define ADC_CS_START_ONCE_BITS
_u(0x00000004)
#define ADC_CS_READY_LSB _u(8)
#define ADC_RESULT_OFFSET _u(0x00000004)
```

There is also a `.EQU` statement to provide a symbolic define for the ADC port that the temperature sensor is connected to:

```
.EQU TEMPADC, 4
```


Step into the `initTempSensor()` routine and gdb switches to displaying assembly language from `adctemp.S`.

```
(gdb) s
initTempSensor () at /home/pi/pico/tutorial5/adctemp.S:22
22 MOV R1, #TEMPADC
```

The first statement loads the ADC port number into register `R1`. Enter a `step` command to perform this.

```
(gdb) s
23 LSL R1, #ADC_CS_AINSEL_LSB
```

However, the ADC's control and status register perform several things and each function occupies a small set of bits within the 32-bit memory location.

“ A loop is performed to wait for the ADC to finish performing this operation ”

This is typical of hardware registers mapped to memory locations. Shift the port number into the correct position. Fortunately, `adc.h` provides defines to help and prevent using an error prone hard coding of the constant. Execute the left shift logical instruction to shift the port number into the correct position with a `step` command.

```
(gdb) s
24 ADD R1, #(ADC_CS_TS_EN_BITS+ADC_CS_EN_BITS)
```

Two other bits in the status and control register control whether the device is enabled. By default, most devices on RP2040 are disabled, to save power; it's up to the user to enable a device before using it. The `ADD` instruction adds the two bits to `R1`. The value of `R1` follows, then `step` through this instruction:

```
(gdb) i r r1
r1          0x4003          16387
(gdb) s
25 LDR R2, adcbase
```

We've formed the value to set into the control and status register. Now load the memory address of this register into `R2` with an `LDR` instruction. Store our value in `R1` into this memory address

using an `STR` instruction. We use an index from the base address to get the correct memory address for the register. Step through these two instructions.

```
(gdb) s
26 STR R1, [R2, #ADC_CS_OFFSET]
(gdb) s
notReady2 () at /home/pi/pico/tutorial5/adctemp.S:30
30 notReady2:LDR R1, [R2, #ADC_CS_OFFSET]
```

The CPU is fast, and hardware devices often take a longer time to perform their operations. A loop is performed to wait for the ADC to finish performing this operation. There is a ready bit in the control and status register which we need to wait to be set. Next, read the value of the hardware register into `R1`, then load 1 into `R3` and shift it into the correct location. Next, perform `AND R1` and `R3` and if the result is zero, then loop to try again.

Single-stepping in the debugger provides plenty of time for the operation to complete, so the loop is not taken and `initTempSensor()` is exited with a `BX` instruction. Single-step through these instructions to exit this routine.

```
(gdb) s
31 MOV R3, #1
(gdb) s
32 LSL R3, #ADC_CS_READY_LSB
(gdb) s
33 AND R1, R3
(gdb) i r r1 r3
r1          0x4103          16643
r3          0x100           256
(gdb) s
34 BEQ notReady2 @ not ready, branch
(gdb) s
35 BX LR
(gdb) s
main () at /home/pi/pico/tutorial5/main.c:26
26 adctemp = readTemp();
(gdb)
```

07 Read the raw temperature value

The routine `initTempSensor()` configured and enabled the temperature sensor, so now it can be read. To read a temperature value, the program tells the ADC to read the temperature sensor, then waits for this to complete, then reads the actual value. Step into the `readTemp()` routine.


```
(gdb) s
readTemp () at /home/pi/pico/tutorial15/
adctemp.S:44
44 LDR R2, adcbase
```

First, load **R2** with the base memory address of the ADC hardware registers. Step through this instruction.

```
(gdb) s
45 LDR R1, [R2, #ADC_CS_OFFSET] @ load
status register
```

Next, execute an **LDR** instruction to load the current value of the status and control register. Now, **ADD** the bit to instruct the ADC to read the temperature sensors value and write that value with a **STR** instruction. Step through these three instructions.

```
(gdb) s
46 ADD R1, #ADC_CS_START_ONCE_BITS @ add
read value once
(gdb) s
47 STR R1, [R2, #ADC_CS_OFFSET] @ write
to do it
(gdb) s
notReady () at /home/pi/pico/tutorial15/
adctemp.S:48
48 notReady:LDR R1, [R2, #ADC_CS_OFFSET]
@ wait for read to complete
```

The loop to wait for this to complete is identical to the one in `initTempSensor()`. Step through the loop with five **step** commands.

```
(gdb) s
49 MOV R3, #1
(gdb) s
50 LSL R3, #ADC_CS_READY_LSB @ done
yet?
(gdb) s
51 AND R1, R3
(gdb) s
52 BEQ notReady
(gdb) s
53 LDR R0, [R2, #ADC_RESULT_OFFSET] @ read
result
```

Finally, read the value. This is simple: we read the ADC results hardware register with an **LDR** instruction. Load the value into **R0**, the register used to return a value to a calling function, then return control to the C code with a **BX** instruction.

Examine the result using an **info register** command. Execute a **step** command to read the value. Execute **i r r0** to display the raw value, then **step** to return.

```
(gdb) s
54 BX LR @ return value
(gdb) i r r0
r0 0x385 901
(gdb) s
main () at /home/pi/pico/tutorial15/main.c:27
27 temp = 437 - (100 *adctemp) / 215;
```

This returns to the C code where we can continue execution or single-step. If continuing to single-step, you will step into the RP2040 SDK's division routine, which uses the RP2040's division coprocessor to perform the division by 215.

08 Modify the program

Congratulations! Welcome to the world of directly programming hardware connected to an ARM CPU. As you **step** through the program,

CMakeLists.txt

> Language: **CMake**

```
001. cmake_minimum_required(VERSION 3.13)
002.
003. include(pico_sdk_import.cmake)
004. project(temperature C CXX ASM)
005.
006. set(CMAKE_C_STANDARD 11)
007. set(CMAKE_CXX_STANDARD 17)
008.
009. pico_sdk_init()
010.
011. include_directories(${CMAKE_SOURCE_DIR})
012.
013. add_executable(temperature
014.   main.c adctemp.S
015. )
016.
017. pico_enable_stdio_uart(temperature 1)
018. pico_enable_stdio_usb(temperature 0)
019.
020. pico_add_extra_outputs(temperature)
021.
022. target_link_libraries(temperature pico_stdlib)
```


adctemp.S

► Language: **Assembly Language**

**DOWNLOAD
THE FULL CODE:**



magpi.cc/learnassembly5

```

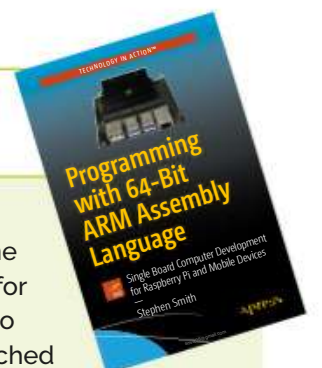
001. @
002. @ Module to interface to the RP2040 ADC
    controller
003. @ as well as the built-in analog temperature
    sensor.
004. @
005.
006. #include "hardware/regs/addressmap.h"
007. #include "hardware/regs/adc.h"
008.
009. .EQU    TEMPADC, 4
010.
011. .thumb_func
012. .global initTempSensor, readTemp
013.
014. @ Initialize the ADC and temperature sensor.
015. @ No input parameters or return values.
016. @ Registers used: R1, R2, R3
    initTempSensor:
017. @ Turn on ADC and Temperature Sensor
018. @ We set the bits to enable the ADC, the temp
    sensor
019. @ and select ADC line 4 (tempadc). All these bits
    are
020. @ in the ADC status register.
021.         MOV     R1, #TEMPADC
022.         LSL     R1, #ADC_CS_AINSEL_LSB
023.         ADD     R1, #(ADC_CS_TS_EN_BITS+ADC_CS_
024. EN_BITS)
025.         LDR     R2, adcbase
026.         STR     R1, [R2, #ADC_CS_OFFSET]
027.
028. @ It takes a few cycles for these to start up, so
    wait
029. @ for the status register to say it is ready.
030. notReady2: LDR     R1, [R2, #ADC_CS_OFFSET]
031.         MOV     R3, #1
032.         LSL     R3, #ADC_CS_READY_LSB
033.         AND     R1, R3
034.         BEQ     notReady2      @ not ready,
    branch
035.         BX      LR
036.
037. @ Function to read the temperature raw value.
038. @ Inputs - none
039. @ Outputs:      R0 - the raw ADC temperature
    value
040. @ Function requests a reading from the status
    reguiter
041. @ then waits for it to complete, then reads
    and returns
042. @ the value.
043. readTemp:
044.         LDR     R2, adcbase
045.         LDR     R1, [R2, #ADC_CS_OFFSET] @
    load status register
046.         ADD     R1, #ADC_CS_START_ONCE_BITS @
    add read value once
047.         STR     R1, [R2, #ADC_CS_OFFSET] @
    write to do it
048. notReady: LDR     R1, [R2, #ADC_CS_OFFSET] @
    wait for read to complete
049.         MOV     R3, #1
050.         LSL     R3, #ADC_CS_READY_LSB
    @ done yet?
051.         AND     R1, R3
052.         BEQ     notReady
053.         LDR     R0, [R2, #ADC_RESULT_OFFSET]
    @ read result
054.         BX      LR
    @ return value
055.
056.         .align 4
057. adcbase: .word    ADC_BASE @ base for analog to
    digital

```

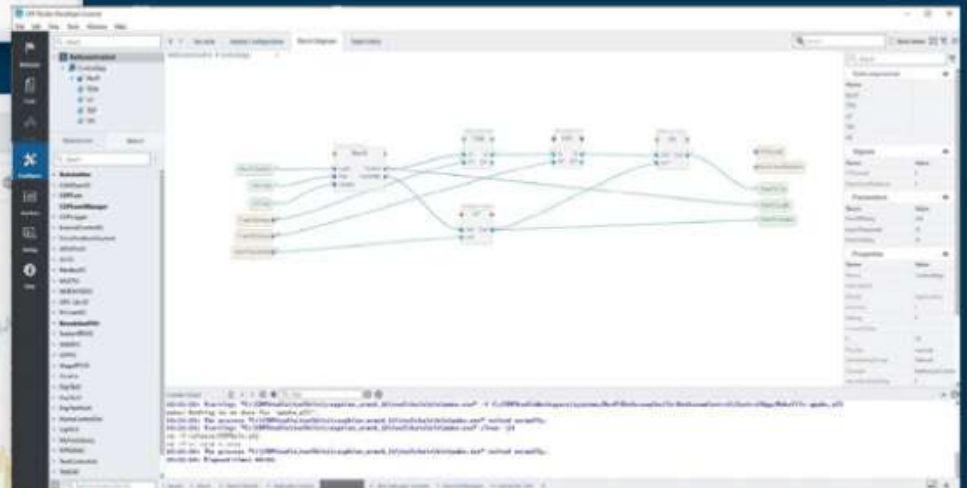
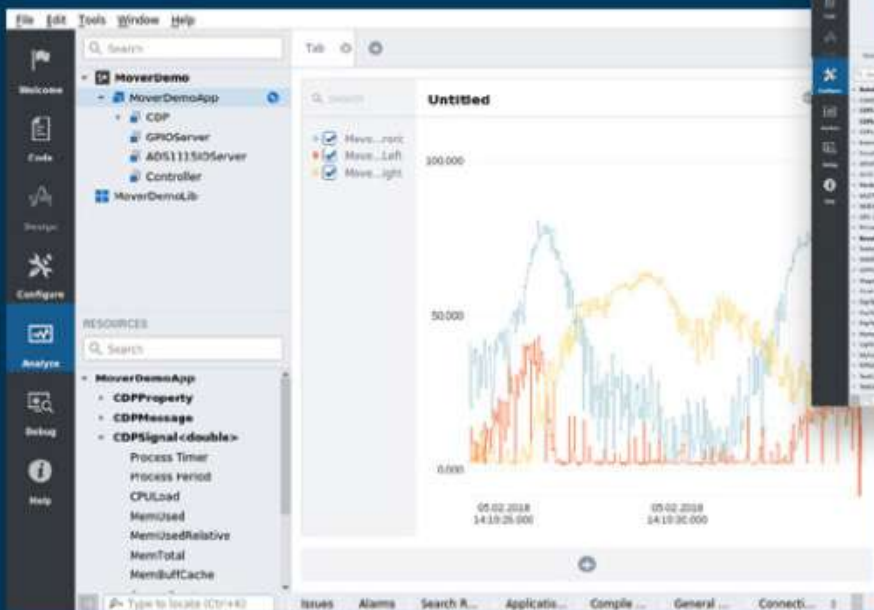
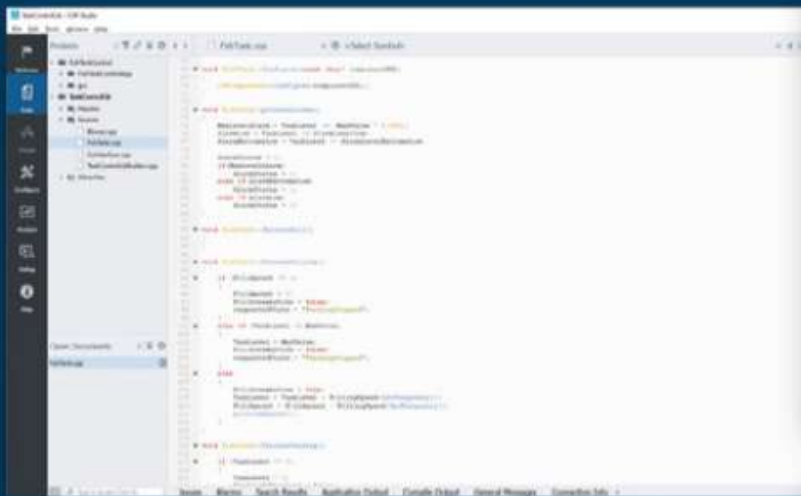
frequently check the values of the registers. The best way to learn programming is by doing. For an exercise, change the program to report the temperature in degrees Fahrenheit, or degrees Kelvin, instead of Celsius. 

Stephen's stuff

He's written three books on Assembly Language Programming. The most recent one is *RP2040 Assembly Language Programming* for the Raspberry Pi Pico, which is the place to go for a deeper understanding of the topics touched on in this tutorial. The first one is *Raspberry Pi Assembly Language Programming* for 32-bit ARM code, and the second is *Programming with 64-Bit ARM Assembly Language*.



Code



Configure

Analyze

With RPi 4 and
Bullseye support

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

CDP Studio, a great software development tool for your home projects. Build systems for Raspberry Pi, use C++ or NoCode programming, open source libraries, out of the box support for GPIO, I2C, MQTT, OPC UA and more. Create beautiful user interfaces. Built for industrial control system development, **FREE for home projects.**

cdpstudio.com

Tel: +47 990 80 900 • info@cdptech.com

CDP Technologies AS // Hundsværgata 8, 6008 Ålesund, Norway



BACK TO SCHOOL 2022

Be Prepared. By Rob Zwartslot

'New year, new you' usually applies to the calendar New Year, but there's no reason not to use a new school year as an excuse to update your at-home workspace in preparation for the learning ahead. Especially when it comes to getting yourself set up to code and make at home.

If you're really keen, you can get a bit of a head start on the computing homework as well. Let's get some kit in and learn how to use it.

Essential kit

Electronics you can keep at home

Raspberry Pi 4 Desktop Kit

| From £110 / \$132

An all-in-one Raspberry Pi starter kit, with a keyboard, mouse, power supply, cables, and a special *Beginner's Guide* book.

magpi.cc/desktopkit



Discovery Kit for Raspberry Pi Pico

| £12 / \$14

An electronics starter kit that will work with both Raspberry Pi and Raspberry Pi Pico! Includes LEDs, resistors, jumper cables, a breadboard, and more.

magpi.cc/discoverykit



CamJam EduKit #3

| £18 / \$22

A very simple robotics kit that is both really fun to build, and a great way to learn more about Python programming.

magpi.cc/edukit3



Soldering iron

| £30 / \$36

You can get cheap kits that have solder, cases, soldering iron holders, etc. on most online stores, but the included soldering irons usually aren't that great. We recommend one of these trusty beginner soldering irons instead!

magpi.cc/antexiron



Project Box 1

| £12 / \$14

With this little kit you can complete ten different Raspberry Pi projects! There are several other project boxes like this that you can find at monkmakes.com.

magpi.cc/projectbox1



Storage

| £35 / \$42

This stylish desktop tidy will help you keep your electronics and other Raspberry Pi bits and pieces neatly tidied away.

magpi.cc/bamboodrawers



If you are having trouble finding a Raspberry Pi for sale, check vpilocator.com

Coding at home

Need to do some computing homework? Let's make that easier

Coding on Raspberry Pi

01 Set up Raspberry Pi

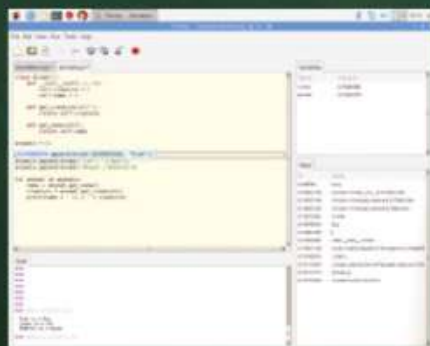
Follow the instructions on magpi.cc/imager on how to download Raspberry Pi OS and get it installed on an SD card. Slot the SD card in and then hook your Raspberry Pi up to a monitor, keyboard, and mouse, before plugging in the power.

During the initial setup, you'll be asked to enter any wireless network details, create a new password, and update the system. Once that's done, head to Raspberry Pi Configuration in the menu and make sure VNC is turned on in the interfaces tab – we'll need that later.



02 IDE for you

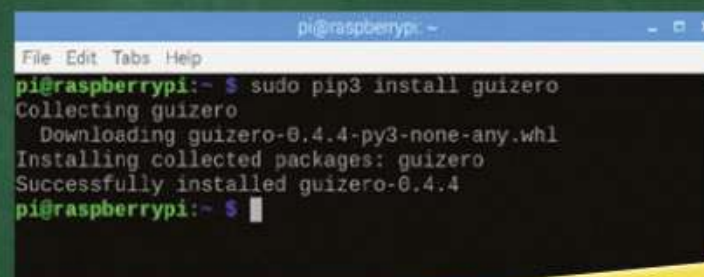
On Raspberry Pi, Thonny is the preferred Python Integrated Development Environment, or IDE. This is a code editor that helps you write a program, run it, and also step through it for bug fixing. Other kinds of IDEs are installed, such as the standard IDLE, so you have several to choose from. You can even install popular third-party IDEs from the Add/Remove Software option in the menu.



03 Installing libraries

Some modules and libraries that you might want to import will have to be installed separately. Usually, you'll have to use the package installer for Python, also known as pip. This will require you to open the Terminal and use something like:

```
pip install pygame
```



Remotely access Raspberry Pi

If you have a limited amount of desk space and no spare monitors and inputs, it might be a hassle to get your Raspberry Pi plugged in each time you want to do some coding. Raspberry Pi OS allows you to dial into your Raspberry Pi via VNC from another computer, allowing you to access the entire Raspberry Pi OS from a window like it's a virtual monitor.

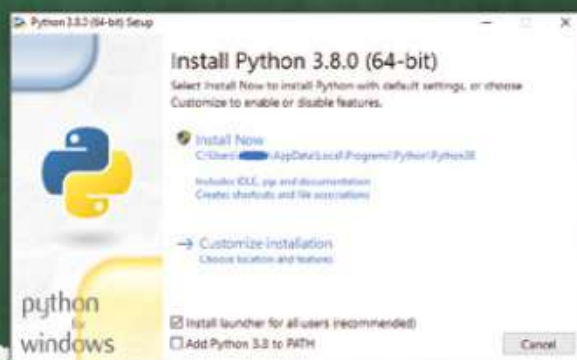
VNC Viewer from realvnc.com is software that works on PCs, Macs, and even Linux computers, so that you can access Raspberry Pi OS. You just need to remember the username and password you set up on Raspberry Pi OS, and the name of the Raspberry Pi, to access it.



Coding on PC

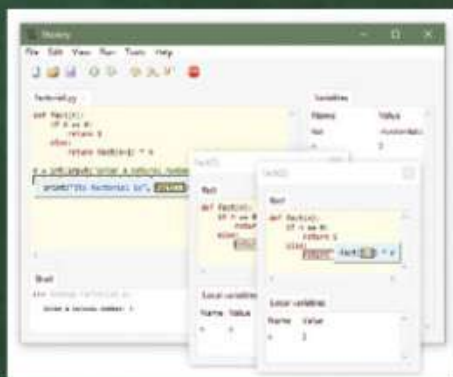
01 Install Python

A PC will not have the ability to run Python scripts by default (along with many other coding languages), so you will have to download and install it. Head to python.org, and you will be able to get the latest version of Python 3. Follow the install wizard, and you're ready!



02 Choosing your IDE

Installing Python will also install IDLE, the default IDE. However, many other programs can be used to write in Python. We quite like Notepad++ (magpi.cc/notepadplus) on Windows. However, Thonny – the IDE on Raspberry Pi – is also available for PC and Mac from thonny.org.



03 Installing libraries

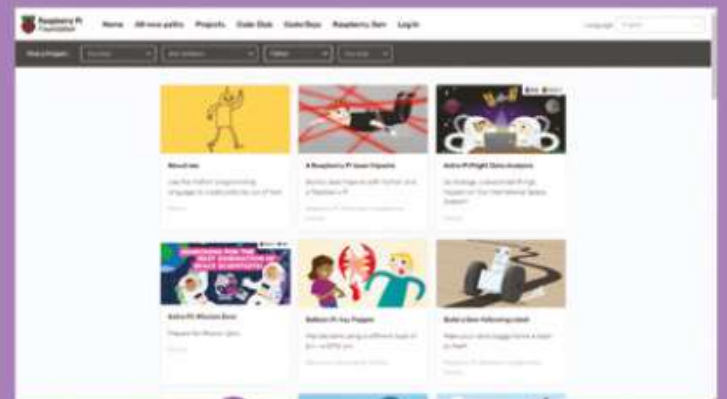
Pip is not installed with the rest of Python, so you'll have to install it manually. In Windows, press the Windows key and type **cmd**, then press **ENTER**. This will open the command line, the Windows equivalent of the Terminal. You can then install pip using:

```
py -m ensurepip --upgrade
```

You can then install packages using pip, just like on Raspberry Pi.

Resources for beginners

Take your first steps into coding



The Raspberry Pi Foundation Projects

Official projects from the Foundation that cover a wide range of topics, from Python to Minecraft hacking, and even 3D modelling with Blender!

magpi.cc/projects

Simple Electronics with GPIO Zero

GPIO Zero is a Python library specifically made for Raspberry Pi that lets you easily connect to buttons, LEDs, and other kinds of inputs and outputs. This book covers how exactly you can use it.

magpi.cc/simpleelectronics



BBC Bitesize

This revision website from the BBC has lots of helpful guides for coding that follows along with the school curriculum.

bbc.co.uk/bitesize



Learn ahead

Get a head start on computing with these projects and resources!

GPIO music box

Python coding, circuit building, game-adjacent programming

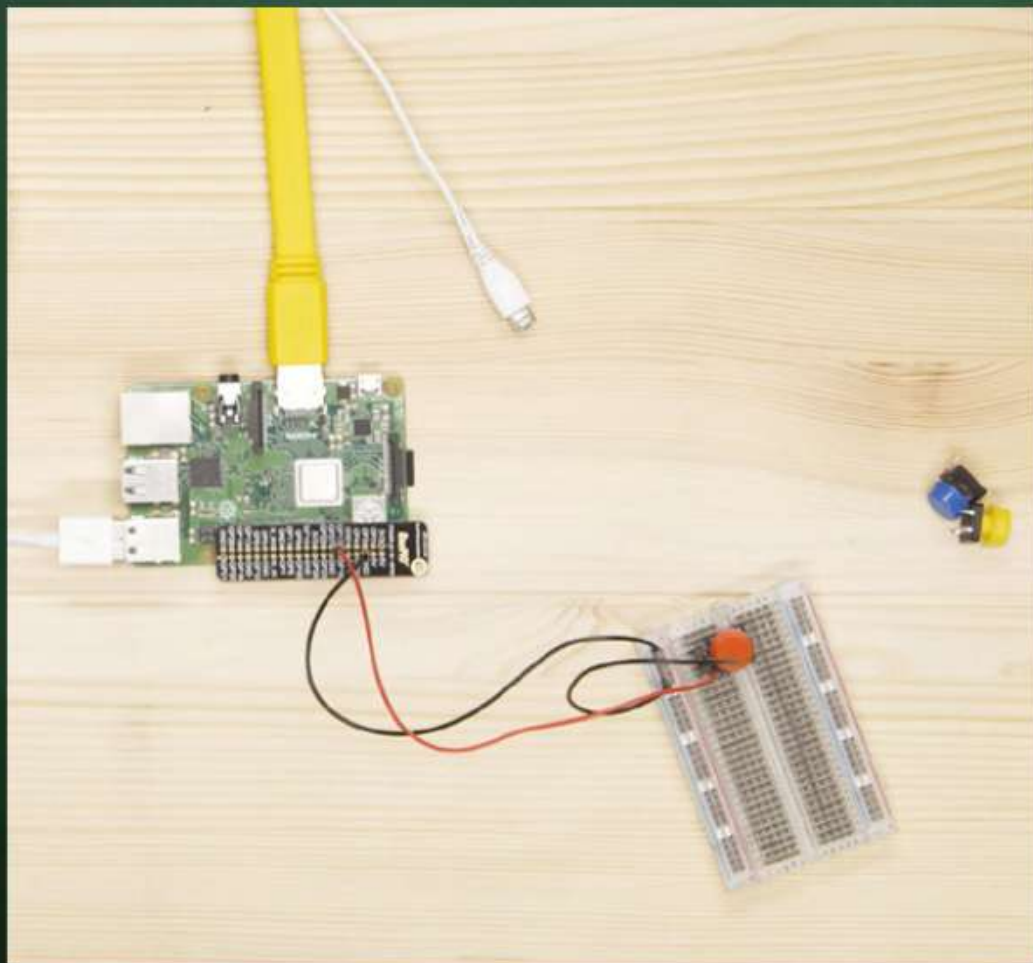
This project is both simple and fun, and teaches about using button inputs with Python, and even Pygame Zero, which can be used for some video game creation.

It doesn't require many components – only buttons, wires, and jumper cables – and can be easily repurposed to do a lot more than just make sounds on a button press.

The ability to play sounds allows you to learn about calling on files from within the code, and how different libraries that can be imported will allow you to skip steps like loops and if statements.

▼ It won't take up much of your desk space

magpi.cc/gpiomusicbox



Whoopi cushion

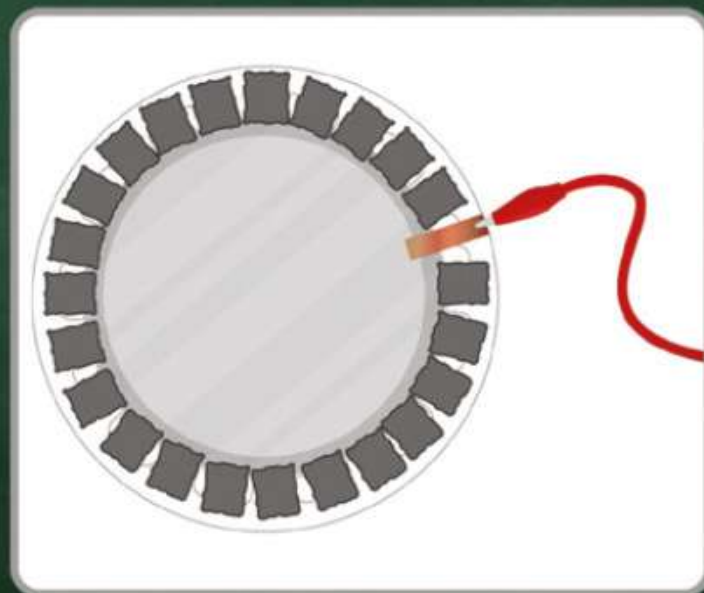
Making, Python coding

One of the best things about this project is that you get to create a custom device that will act as a button/whoopee cushion out of household items like paper plates and paper clips.

It works in a similar way to the music box – a button will activate a sound – however, in this case, it uses a loop to check for when the button is pressed. Also, you can trick someone to sit on it, which is pretty funny.

It also gives you a look into GPIO Zero, which is a really easy way to connect inputs and outputs to Raspberry Pi via code.

magpi.cc/whoopi



More advanced resources

Take your first steps into coding

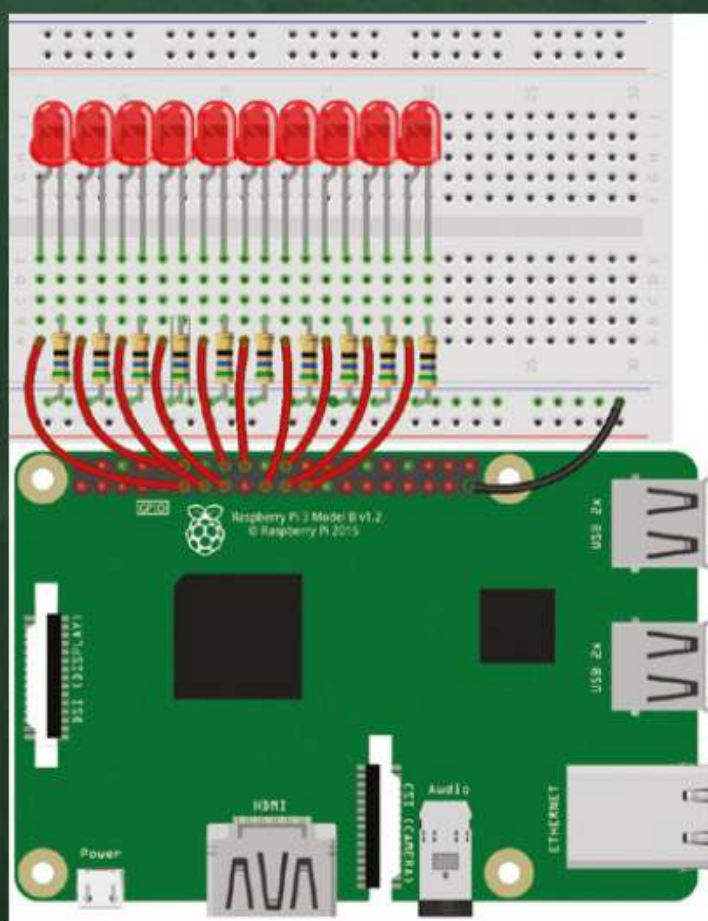
People in space

Using online data, storing and sorting data, LEDs

This simple-looking project hides a much deeper use of code than explored in the previous projects. This one makes use of a web API, which is a kind of service linked to a data source that you can interact with using code. In this case, it involves finding out who is on the International Space Station.

With extra data comes a new way to sort and read the data, which is done using lists and dictionaries in the code here. It's a big step up from the previous tutorials, but it's still simple enough so you can learn what each part is.

magpi.cc/peopleinspace

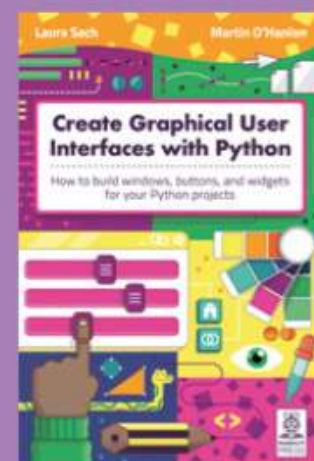


◀ The circuit for this is a lot of LEDs, but they're easy to arrange

Create Graphical User Interfaces with Python

Want to make software you can use on a desktop? This book will help you through learning the ins and outs of graphical interfaces, while also expanding your other skills.

magpi.cc/createguis



Wearable Tech Projects

This book is a great guide to the kinds of things you can do with electronics, and how you can wear them with pride as well.

magpi.cc/wearabletech



Code the Classics

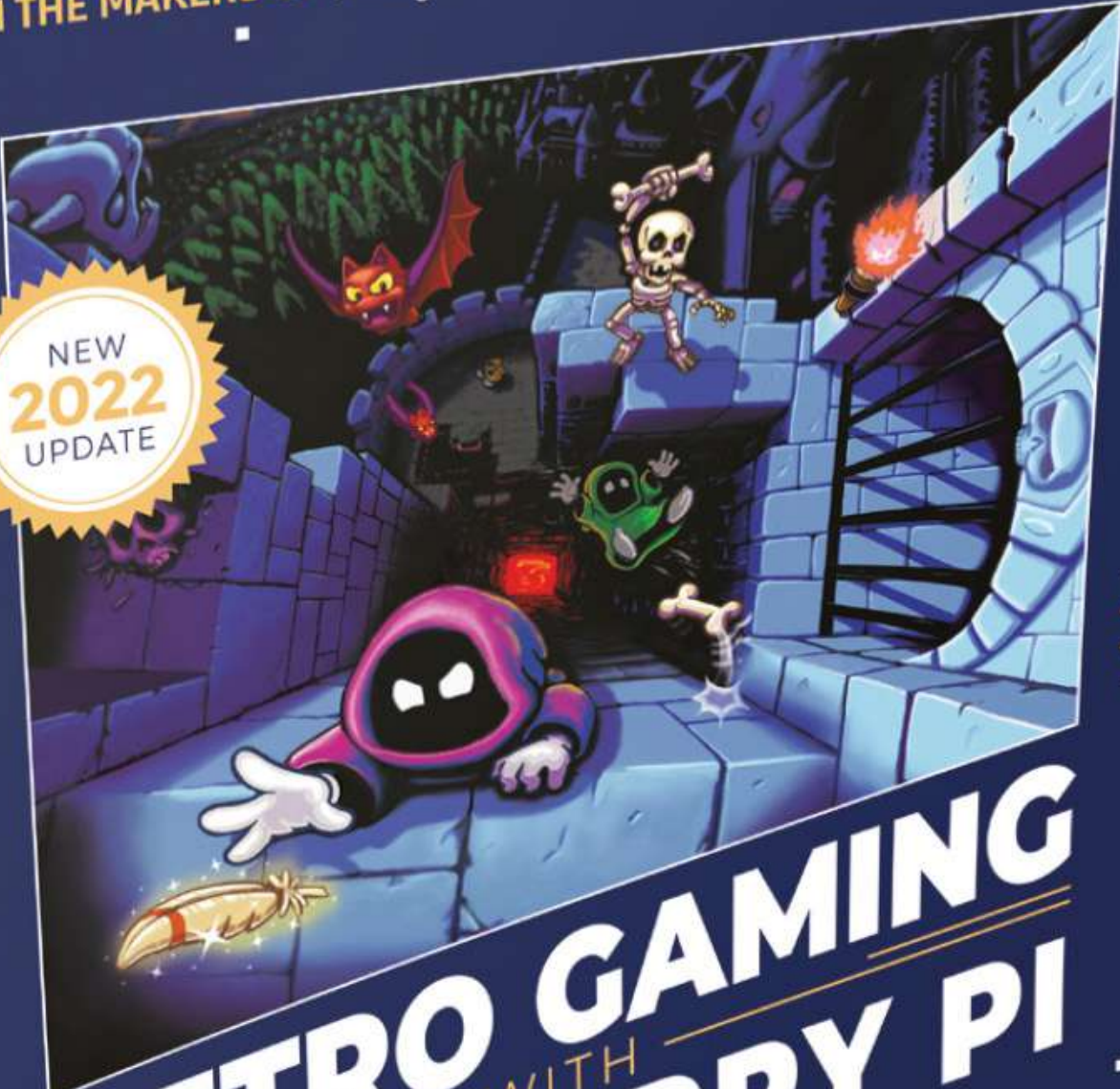
If you've ever dreamed about making your own games, you can't go far wrong with *Code the Classics*, which uses Python to recreate many older games.

magpi.cc/codetheclassics



FROM THE MAKERS OF *MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

NEW
2022
UPDATE



**PLAY
& CODE**
GAMES!

RETRO GAMING WITH RASPBERRY PI

2ND EDITION

164 PAGES OF
VIDEO GAME PROJECTS



RETRO GAMING

WITH

RASPBERRY PI

2ND EDITION

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games. Build your own games console or full-size arcade cabinet, install emulation software and download classic arcade games with our step-by-step guides. Want to make games? Learn how to code your own with Python and Pygame Zero.

- ***Set up Raspberry Pi for retro gaming***
- *Emulate classic computers and consoles*
- ***Learn to code your own retro-style games***
- *Build a console, handheld, and full-size arcade machine*



BUY ONLINE: magpi.cc/store

CrowPi L

SPECS

DIMENSIONS:

Size: 291 × 190 × 46 mm (L×W×H);
Weight 1.1 kg

INPUT/

OUTPUT:

11.6-inch
1366×768
IPS screen;
2-megapixel
camera with
microphone;
3.5 mm
headphone jack;
USB keyboard;
touchpad;
stereo speakers

POWER:

5000 mAh
battery
(approximately
three hours of
charge time/
use); DC 12V
2A adapter;
USB-C interface

► Elecrow ► magpi.cc/crowpil ► From £169 / \$203

Elecrow laptop turns Raspberry Pi 4 into a battery-powered laptop and electronics learning kit. By **Lucy Hattersley**

CrowPi L is a new laptop from Elecrow. It builds on top of the previous CrowPi2 build that we tested in *The MagPi* issue #97 (magpi.cc/97).

It's billed as "a lite version born out of CrowPi2" that reduces the size of the laptop and adds a battery.

The electronics kit is now bundled alongside the laptop instead of fitted beneath the keyboard.

You will need to source your own Raspberry Pi 4. Starting from \$203 without the Crowtail kit is good value; even \$250 for the CrowPi L and Crowtail electronics kit works out cheaper than its predecessor. Factor in around \$50 for shipping.

Design matters

The design is a step forward. The white clamshell case features an 11.6-inch screen, chiclet-style keyboard, and a small touchpad repositioned in the top-right. Included is a 2.4GHz wireless mouse. Poor touchpads plague Raspberry Pi laptops; CrowPi L solves this problem by pushing the touchpad out of the way.

The internal design is clever: you attach four magnets to Raspberry Pi and HDMI expansion board and a 2-in-1 TF card adapter (an A/B switch on this board enables you to swap between two operating systems).

Two ribbon cables connect to Raspberry Pi's USB port on Raspberry Pi and bridge between the HDMI expansion board and CrowPi L's motherboard.

Be aware that we needed a Torx T5 screwdriver to attach the magnets.

Removing a separate panel with a Torx T6 screwdriver reveals the 5000 mAh battery that provides CrowPi L with approximately three hours of runtime.

In the box was a 32GB card with CrowPi's custom operating system based on Raspberry Pi OS (Debian Buster). We also tested out Raspberry Pi OS on the second drive (and via USB boot using an M.2 drive; and Batocera.linux for retro gaming).

Expansion

The GPIO pins are broken out via a smaller 1.2 mm pitch 40-pin socket. If you want to use regular HAT hardware with CrowPi L, you'll need the 2.54 mm CrowPi L GPIO Breakout board, available from Elecrow for \$2 (magpi.cc/crowpilbreakout).

Ethernet and USB ports from Raspberry Pi sit on the left; to the right side is a USB-C charging socket, 3.5 mm audio minijack connection, full-size HDMI connector, and the smaller 1.27 mm pitch GPIO connection (that connects to the CrowPi L Base Shield with its 20 JST connectors for quick electronics prototyping).

CrowPi L OS

We used the spare microSD port to test Raspberry Pi OS. It runs fine, although we did lose access to the battery charge menu item. Oddly, our screen

▼ Raspberry Pi 4 is fitted magnetically inside the laptop via a clever design



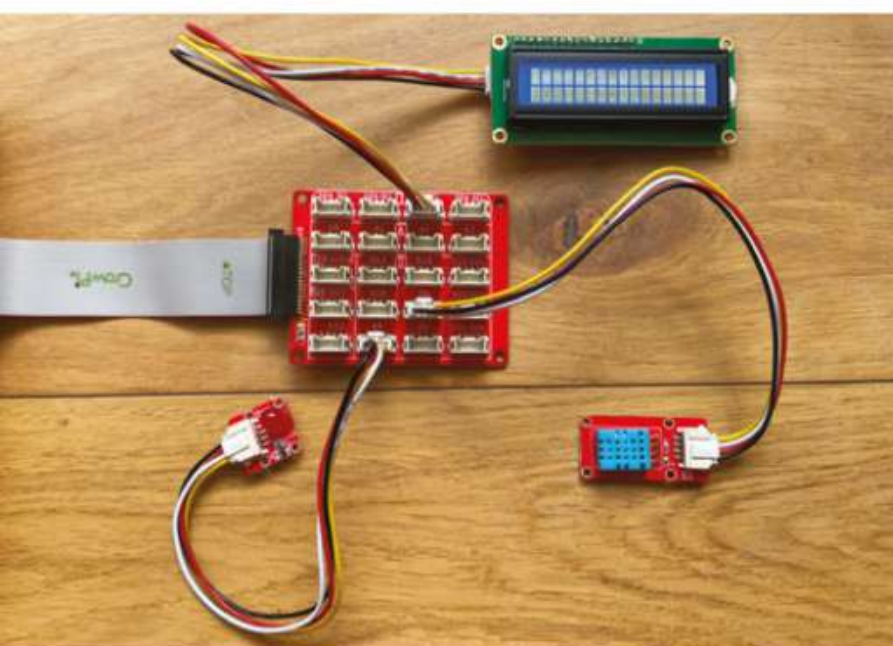


displayed a resolution of 1920×1080 with the stock Raspberry Pi OS, and a look back at config.txt in CrowPi L OS revealed custom timings to set the resolution to 1912×1079.

CrowPi assures us that we have the 1366×768 screen as supplied with all models. We found a resolution of 1280×720 worked best with Raspberry Pi OS.

On the whole, CrowPi L is a nice piece of kit. We're typing up this review on it. It's chunky: measuring 4.5 cm (1 3/4-inch) at the rear and tapering down to 2 cm at the front and, with Raspberry Pi 4 inside it, weighed it in at 1172 g (2.58 lb). Two speakers offer passable sound and a 2MP webcam worked out of the box. We think it'd be perfectly possible to do a day's work on CrowPi L.

▲ The CrowPi L laptop features a great keyboard, small screen, and tiny touchpad



▲ The CrowPi L Base Shield is connected for electronic prototyping

“ We think it'd be perfectly possible to do a day's work on CrowPi L ”

As with the CrowPi 2, the Crowtail Starter Kit elevates this laptop. The electronics kit comes with 22 modules: LCD, micro-speed motor, 9 g servo, battery pack, and a button, buzzer, and sensors, plus an infrared remote control. There's everything you need here to create a vast range of different builds, and a manual walks you through 21 builds, from Hello World to a remote-control door. **M**

Verdict

CrowPi L is fantastic value and it delivers a good laptop and great electronics learning experience at a superb price. Recommended!

9/10

Motor SHIM for Pico

SPECS

MOTOR DRIVER:

DRV8833 dual H-bridge

FEATURES:

2 × JST-ZH 2-pin motor connectors, on-board push-button, Qwiic/STEMMA QT breakout connector

DIMENSIONS:

26 × 21 × 5 mm (inc. headers)

► Pimoroni ► magpi.cc/motorshim ► £10 / \$10

This tiny motor driver board is ideal for making mini robots with Pico. By **Phil King**

Raspberry Pi Pico's diminutive size makes it highly suitable for using in smaller robots, and now it has a mini motor driver board to match. Pimoroni's Motor SHIM is about half the size of Pico and fits onto the ten pins nearest the latter's micro USB port.

You can solder the Motor SHIM straight onto Pico's pins for a permanent fit – orientation is aided by two mounting holes and a silkscreen image on its underside – or use a couple of female

headers soldered onto the SHIM if you want to remove it later.

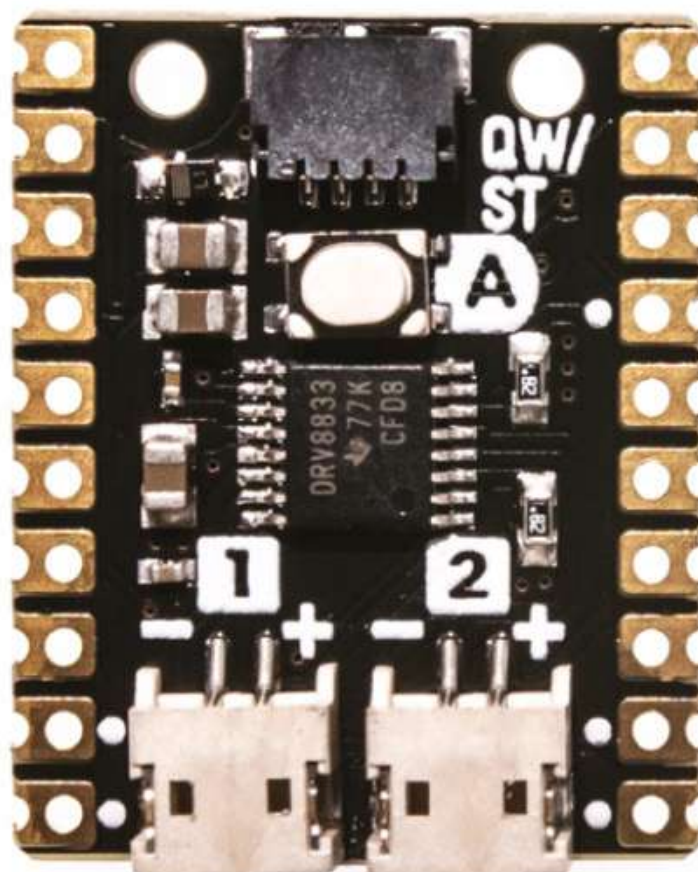
The Motor SHIM features a DRV8833 dual H-bridge motor driver. This means it can drive a maximum of two motors – the rival Kitronik Robotics Board can handle four, but then it is considerably larger.

Rather than screw terminals, the Motor SHIM features two small two-pin JST-ZH connectors. For each, just plug in one end of a cable (not supplied), and the other end into a Motor Connector SHIM on a micro metal-gear motor. You can also buy motors with the JST-ZH connector pre-soldered – to the top or side, depending on your mounting preference. Note that the Motor SHIM won't work with motors equipped with six-pin Micro Metal Motor Encoders, however.

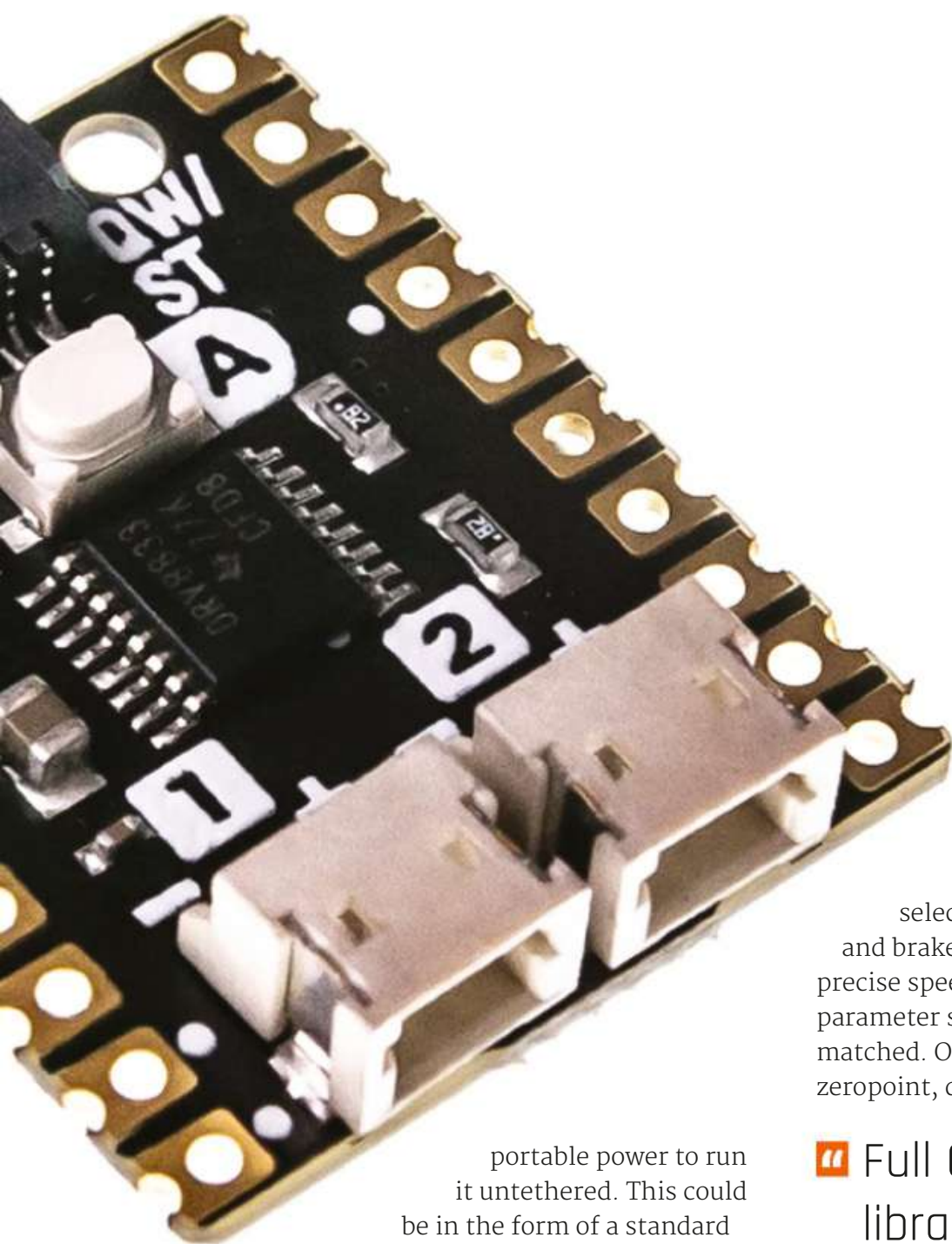
Chassis and power

With your motors connected, you can use brackets to mount them on a two-wheeled robot chassis. The Motor SHIM's product page features the design (DXF file) for a laser-cut mini chassis. Or you could use an existing robot chassis you have to hand – we borrowed one from a Trilobot – or 3D-print one. If you need some inspiration, check out Kevin McAleer's BurgerBot (magpi.cc/burgerbot), which uses Pico and the Motor SHIM.

While you'll connect your robot's Pico via USB to a computer for programming, you'll need



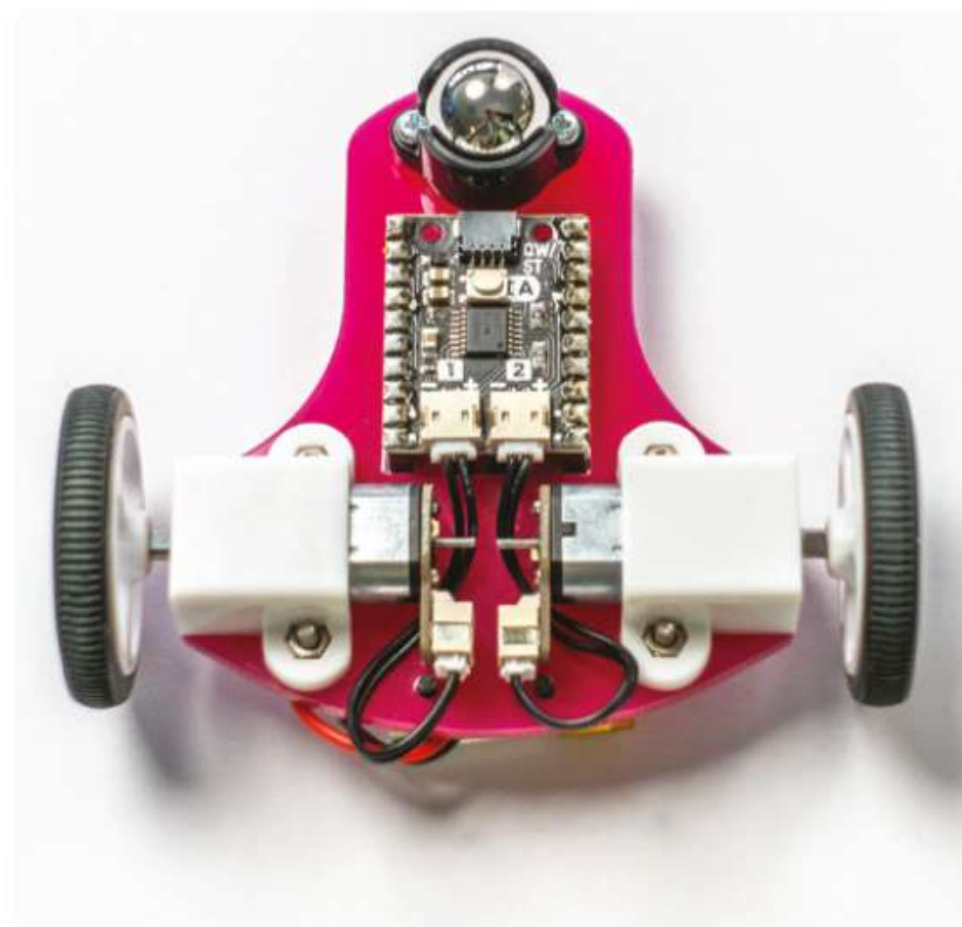
► As well as two JST-ZH motor connectors, there's an on-board push-button and Qwiic/STEMMA QT port



portable power to run it untethered. This could be in the form of a standard USB power bank, but a LiPo battery pack is a more slimline solution. For the latter, you'll require a LiPo SHIM mounted above/below the Motor SHIM using a stacking header, or (less neatly) connecting via a Pico Omnibus Dual Expander.

Easy to program

Full C++ and MicroPython libraries are provided for the Motor SHIM, packed with useful functions to make it easy to control your robot. We did need to reverse the direction of our right-hand motor, although that's easily achieved with a parameter



▲ The Motor SHIM fitted to the base of a simple two-wheeled robot

◀ This tiny board is half the size of a Pico and can drive two motors

when creating its 'motor' object in the code setup.

As well driving one or both motors forward and backwards at a selected speed, there are helpful coast and brake functions. You can calibrate the precise speed of each motor with a 'speed_scale' parameter so that both motors are perfectly matched. Other settings include the deadzone, zeropoint, duty cycle, and frequency. The latter

“ Full C++ and MicroPython libraries are provided for the Motor SHIM, packed with useful functions ”

is even used in one of the code examples to play a tune on the motors!

The software library includes a function to read the Motor SHIM's on-board button, which may prove handy for stopping/starting a movement sequence. You could also use its single Qwiic/STEMMA QT port to add an I2C sensor such as an ultrasonic range-finder to your mini robot. 

Verdict

While you'll need to add motors and a lot of other bits, this tiny Motor SHIM is a great choice for making a pint-sized Pico robot, aided by excellent software libraries.

9/10

HiFiBerry DAC2 Pro & HD

SPECS

DAC:

Dedicated 192kHz/24-bit high-quality Burr-Brown

SIGNAL-TO-NOISE RATIO:
112 dB

SAMPLE RATES:
44.1-192kHz

COMPATIBILITY:
All 40-pin GPIO Raspberry Pi models

▼ The Pro model features an amplified headphone socket and several useful headers

► HiFiBerry ► hifiberry.com ► Pro: £35 / \$42; HD: £90 / \$108

These two DACs from HiFiBerry are a perfect blend of quality and affordability for premium audio. **PJ Evans** turns it up

HiFiBerry was one of the first firms on the scene with a range of DACs, (digital-to-analogue converters) that take Raspberry Pi's modest audio capabilities and puts them right up there with the audiophile best. In Raspberry Pi terms, a DAC is an add-on (HAT) that provides much higher quality audio output than you would normally expect.

A rule of thumb is that the more you pay for these boards, the higher sound quality can be achieved (although the law of diminishing returns most definitely applies). HiFiBerry is offering two products, the DAC2 HD and Pro, that sit in sensible places on the spectrum between the basic sound and the ultimate hi-fi.

On the more affordable end is the DAC2 Pro. This features a dedicated 192kHz/24-bit DAC, low-jitter clocks, and low-noise voltage regulators, all with the purpose of producing the best sound possible at that price point. It also features a headphone amplifier for convenience.

A lot more punch

If you're looking for something a little more special and suitable

for professional use, the DAC2 HD packs a lot more punch by separating out many of the Pro's components into discrete parts, allowing HiFiBerry to source the best quality in all cases.

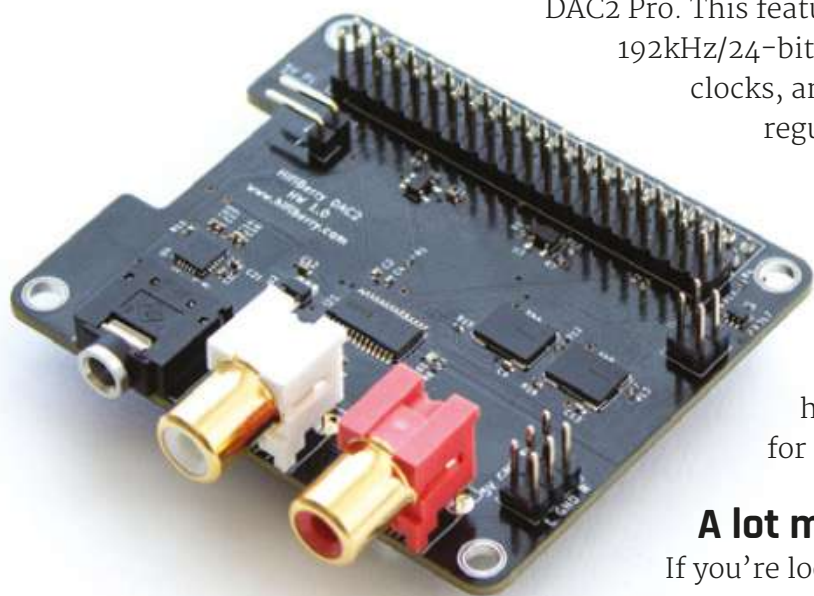
“If you're looking to build a whole-home audio system or something for studio work, these boards are fine choices”

Thoughtfully, HiFiBerry offers a dedicated operating system that makes installation as simple as connecting the DAC and powering up. We instantly had features such as Bluetooth and Apple AirPlay with zero effort. In both cases, the sound was impressive and rich. If you're looking to build a whole-home audio system or something for studio work, these boards are fine choices. **M**

Verdict

Another impressive bit of hardware design from HiFiBerry, offering both choice and quality at sensible prices. Combined with HiFiBerry OS, these boards are ideal for home audio projects.

8/10

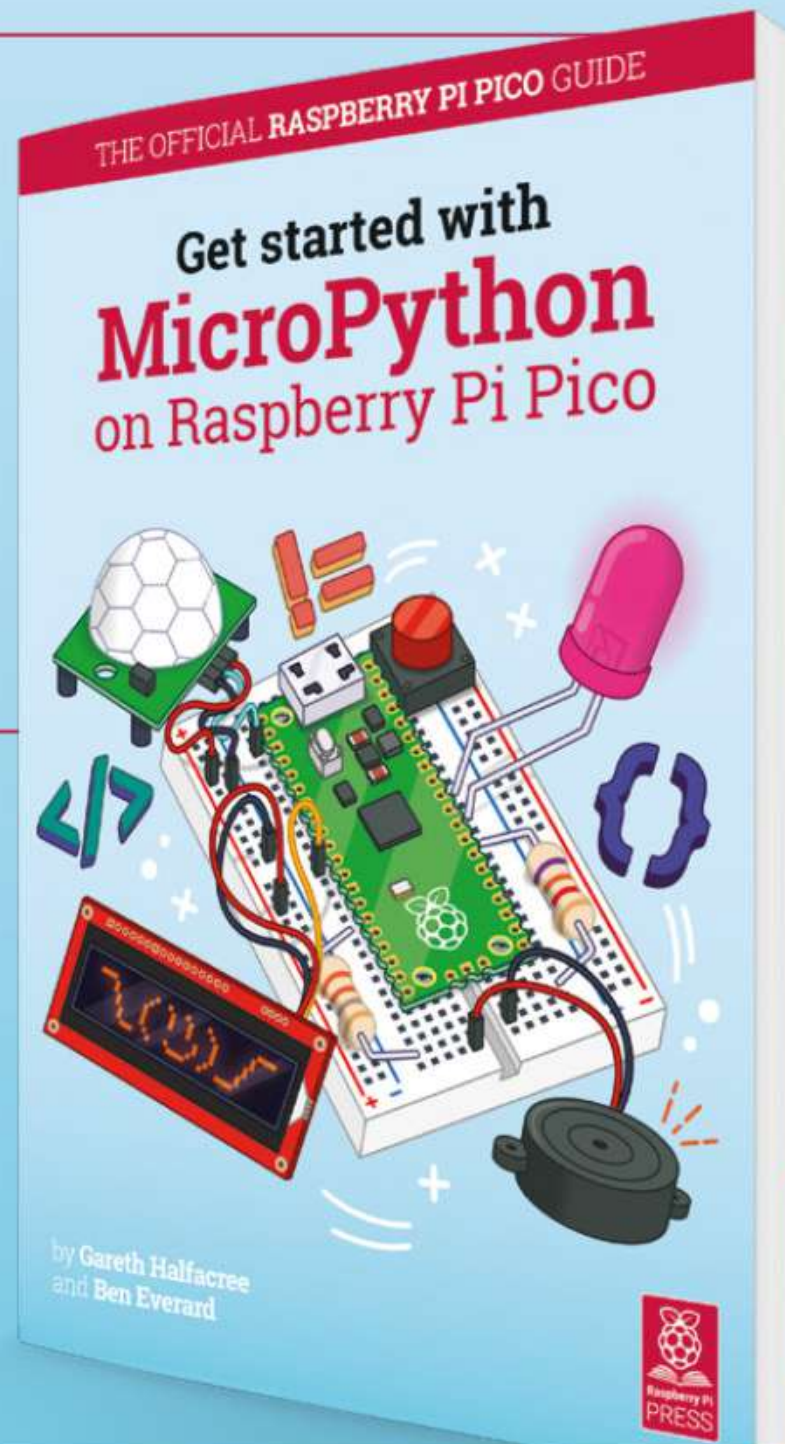




Get started with MicroPython on Raspberry Pi Pico

Learn how to use your new Raspberry Pi Pico microcontroller board and program it using MicroPython. Connect hardware to make your Pico interact with the world around it. Create your own electro-mechanical projects, whether for fun or to make your life easier.

- Set up your Raspberry Pi Pico and start using it
- Start writing programs using MicroPython
- Control and sense electronic components
- Discover how to use Pico's unique Programmable IO



Available now: magpi.cc/picobook

Mini Pupper

SPECS

TECHNOLOGIES:

SLAM (Simultaneous Localization and Mapping), Full Self-Navigation, OpenCV

PHYSICAL SPECS:

209 × 109 × 165 mm, 560 g

ELECTRONICS:

800 mAh battery, 240 × 320 LCD screen, 13 × custom servos

► MangDang ► mangdang.net ► From £335 / \$399

This small robot dog is cute, fun, and great for robotics learning. **Rob Zwetsloot** takes it for walkies

Before the year 2000, we thought robots would be walking around on two legs.

While this may have been because of various logistical issues and metaphors in TV sci-fi, it still didn't quite prepare us for the amount of wheeled robots that have become the standard.

Mini Pupper breaks that mould. While not bipedal like the old Rapiro, it's still great to see a step in the right direction (pun unintended, but it's being left in). Not only does it look like the kind of robot we always expected, it's also well-designed

to act like a robot, in that it has a lot of technology that allows it to be automated.

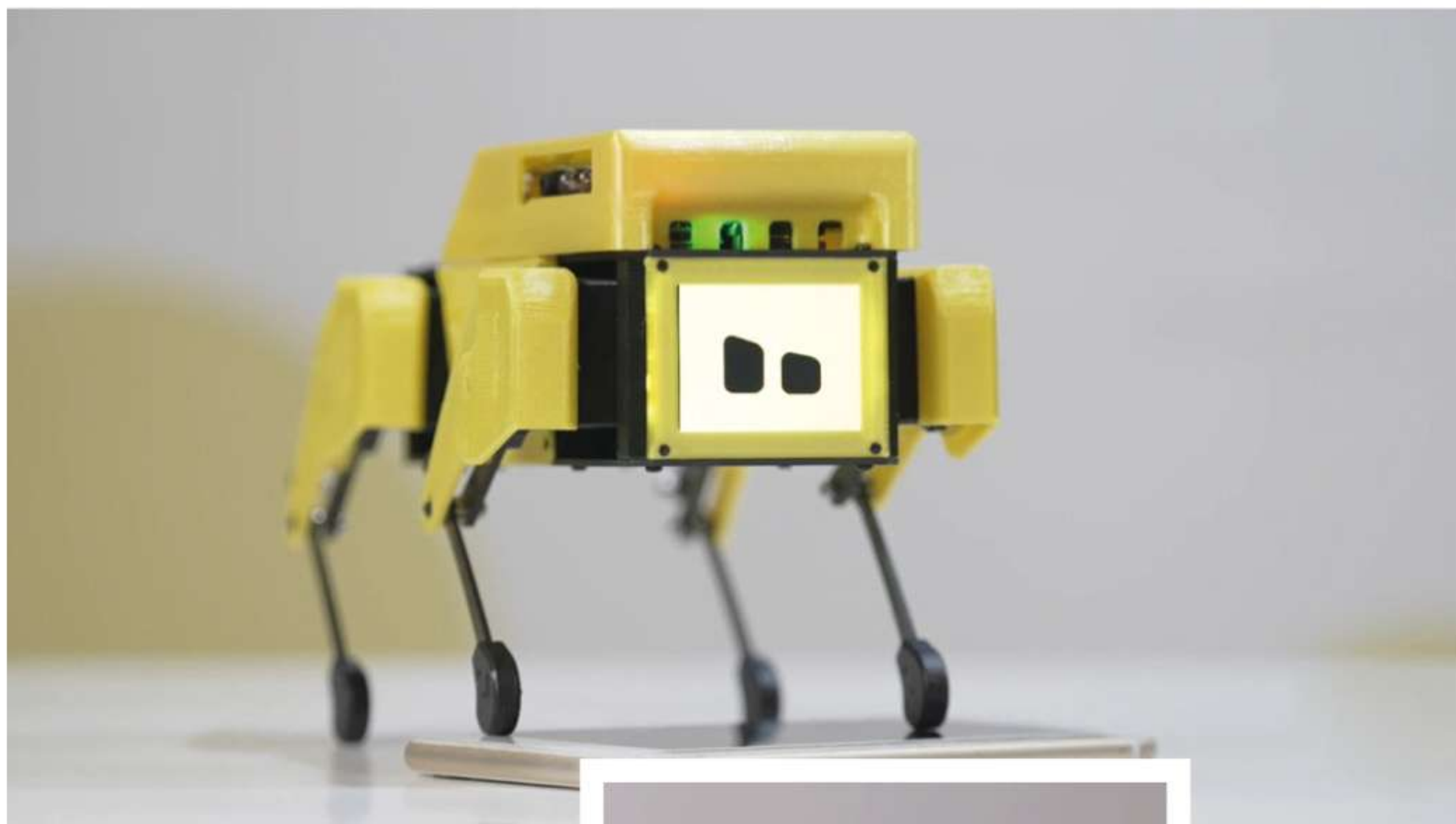
ROS (Robotic Operating System) is the OS installed on Raspberry Pi for Mini Pupper, and Mini Pupper includes OpenCV, SLAM (Simultaneous Localization and Mapping), and other navigation technologies. It will create a map of an area, much like a Roomba so that it can navigate it better.

Pupper in the window

Our Mini Pupper came fully built, however, you do have the option to save some money and supply your own parts, like a Raspberry Pi, so that you can construct it yourself. While the construction methods are constantly being updated and



▲ Three levels of kits are available to build at home



“ Movement of Mini Pupper looks fairly odd, but it can move around just fine on a flat surface ”

improved, it's not a quick build and it is a touch fiddly – it is quite a small kit after all!

Once constructed, the body feels pretty sturdy though, and the legs are much better than they may seem in photos. It weighs just over half a kilo, yet the legs don't really have much of a problem holding it up. Out of the box, you can connect a Bluetooth controller to move around Mini Pupper and it has many movement modes such as trotting, lowering, raising, sideways shuffling, and many more. The makers claim 12 degrees of freedom because of this.

New tricks


Customising and programming Mini Pupper requires you to plug Raspberry Pi into a PC so that you can access ROS onboard. Here, you can start playing around with the facial settings and also get



▲ It is quite an odd looking robot, but it's very adorable in its own way – and compact!

it connected to your wireless network – the latter of which you'll need to make use of the navigation abilities and mapping.

Movement of Mini Pupper looks fairly odd, but it can move around just fine on a flat surface, although we had huge trouble on carpeted floor, so your ability to make it map a room may depend on the floor. As it's for learning and tweaking with code, it will likely mostly live on a tabletop, so we can forgive it for that.

While it is quite expensive, it is very impressive with the amount it can do – thanks to the mixture of parts and ROS running it. It's also fairly expandable as well. 

Verdict

A fun little robot that is a great way to try out several kinds of robotic programming. Might stay on your desk though.

8/10

ONLY THE
BEST

Tindie Treats

A few of our favourite projects found on Tindie, the marketplace for electronics

By Marc de Vinck

 @devinck

Since the last time I wrote about a few of my favourite Tindie products, the company has seen an additional 100,000 orders shipped and 3000+ more products added, shipping to more than 160 countries around the

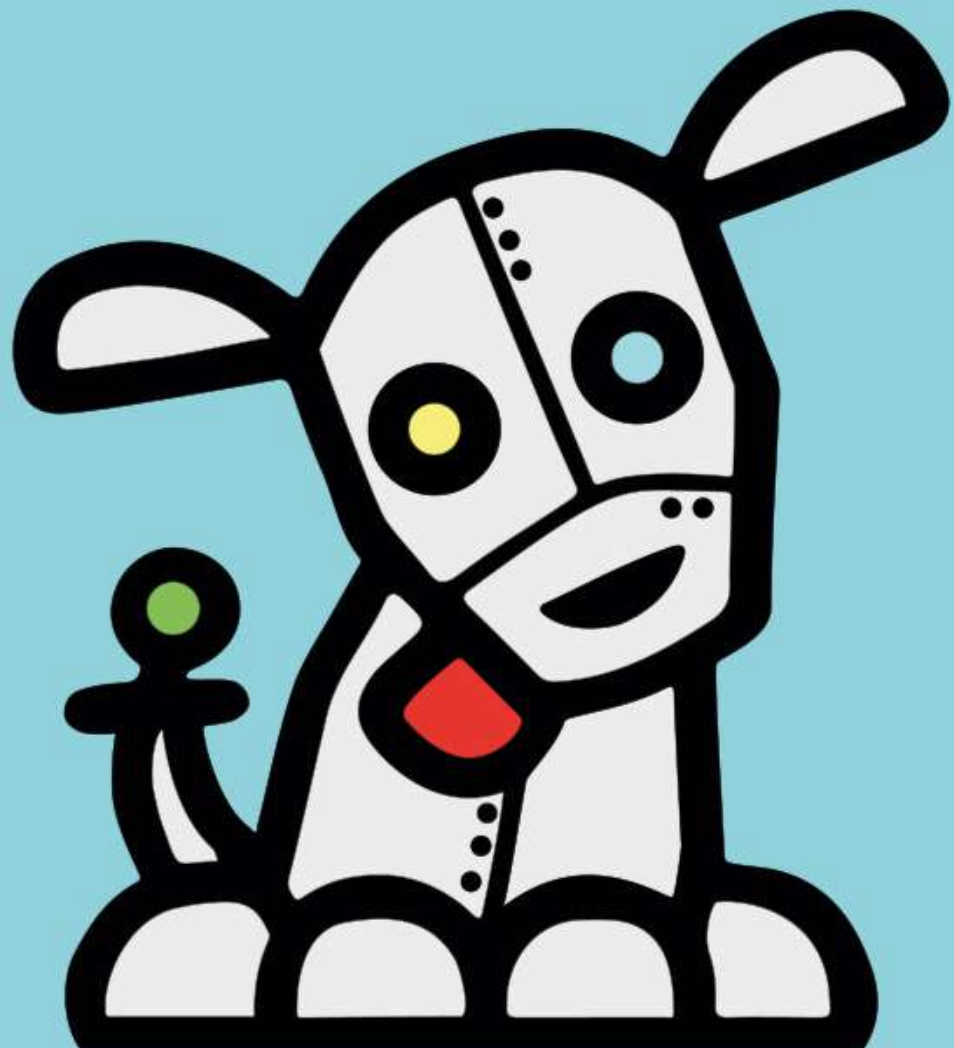
world. If you are unfamiliar with Tindie, well then, I have a treat for you! Tindie is an online marketplace that features mostly the small DIY technology creators. It's a great place to find esoteric kits and components for making unique projects.

I have ordered plenty of kits from Tindie and, so far, so good. In fact, I bought another kit while searching for products for this Best of Breed! The website is organised by topics like 3D printing, IoT, robots, sound, camera, and of course DIY technology, which I thoroughly enjoy browsing on a regular basis. And many of those topics include several subtopics, making browsing the Tindie website easy and fun. And a bit dangerous!



HackSpace

This feature is from HackSpace magazine. Each issue includes a huge variety of maker projects inside and outside of the sphere of Raspberry Pi, and also has amazing tutorials. Find out more at hsmag.cc.



RP2040 Stamp vs CircuitMess Chatter

SOLDER PARTY ♦ \$12 | hsmag.cc/stamp

CIRCUITMESS ♦ \$14.99 | hsmag.cc/diywireless

The RP2040 Stamp, designed by Solder Party in Sweden, is a hand-solderable Raspberry Pi RP2040 IC module with 8MB of flash and LiPo battery charging management, featuring a status LED, a reset button, and a multi-colour NeoPixel LED. By building this module with 2 mm pitch castellated edges, Solder Party has made it easy for you to add an RP2040 to your own PCB without worrying about hand-soldering the extremely small-pitch QFN integrated circuit.

The creators also provide well-documented source files including code, drawings, pinouts, and more. If you want to start prototyping with an RP2040, this is an affordable, time-saving little board. It even comes pre-flashed with CircuitPython 7.1.0-beta!



Left The easy way to build an RP2040 board

VERDICT

RP2040 Stamp

Great, but you'll have to solder!

10/10

CircuitMess Chatter

Useful long-term application.

9/10

Chatter, designed by CircuitMess in Croatia, is a DIY electronics kit for building your own two-way text messaging communicator. The kit

comes with a pair of radios because, as the creator states, "texting yourself is not as fun as you think"! They operate on the radio spectrum commonly used by LoRa devices, which is licence-free and operates in the sub-gigahertz range. It's a bit slow for modern smartphones, but it's great for small chunks of data like a simple message, and it works reliably over long distances.

You can send texts, emojis, memes, and GIFs. When building the kit, you will learn about wireless security, LoRa radios, encryption and decryption, basic code, and how to build simple apps for your communicators. This is a great way to get someone excited about DIY electronics! →



Below An easy introduction to LoRa



Pico Cube

SB COMPONENTS ♦ \$37.99 | hsmag.cc/picocube

3

D LED cubes have been around for a long time. They are a relatively popular kit for DIYers. However, they are also typically difficult to build, as they usually rely on free-form soldering of many through-

hole LEDs to form the cube. I have built a few 3×3×3 cubes, and although they came out well, they weren't exactly fun to put together. I never attempted to build a 4×4×4.

The Pico Cube, from SB Components in the UK, is a 4×4×4 cube that runs via a Raspberry Pi Pico and aims to make building a 3D LED cube actually enjoyable. They took out a lot of the variability of the building process by integrating PCBs instead of relying on the LEDs' wires for the structure. And for those of you that don't want to deal with soldering (gasp!), they even offer a fully assembled version!

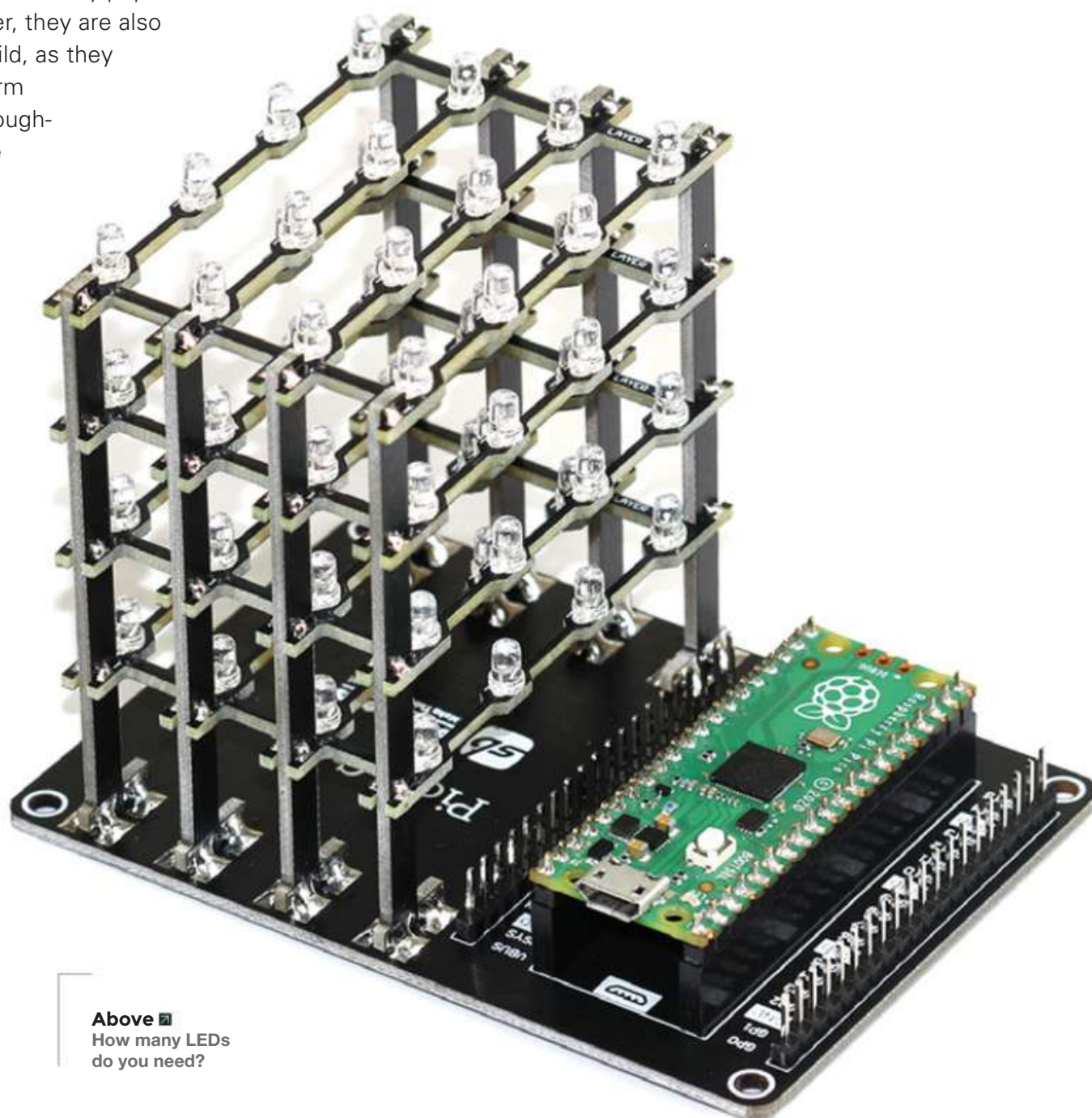
VERDICT

Pico Cube

An easier way to build a cube.

8

/10



Above ■
How many LEDs
do you need?

1.28 inch Round LCD HAT

SB COMPONENTS ♦ \$36.99 | hsmag.cc/roundlcd

The Round LCD HAT for Raspberry Pi, by SB Components, is a 1.28-inch RGB display HAT with 240x240 resolution and a joystick. It uses the standard 40-pin Pico GPIO interface, so it's simple plug and play. OK, plug and code, then play!

The HAT incorporates a display driver and SPI interface, which reduces the required available pins, allowing you to add more components for your project. It also features a four-way joystick with central button. It's a bit of a niche product, which is exactly what I like about the products and creators on Tindie. Something for everyone! →



Above ■
Circles are the new squares

VERDICT

1.28 inch Round LCD HAT

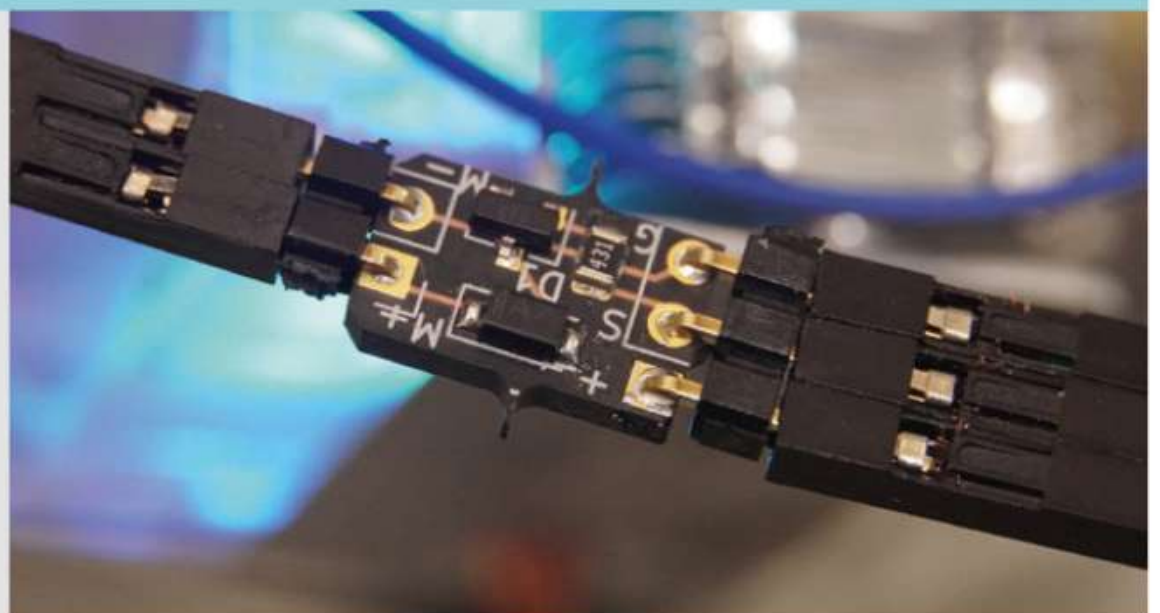
If you want a round display, this might be a good fit.

8/10

EZ FAN2

JC DEVICES ♦ \$8.95 | tindie.com


The Raspberry Pi is a great computing platform. And as the technology advances, and the processor gets faster, managing the heat it produces becomes more important, especially in a small enclosure. With the EZ Fan2 by JC Devices, you can easily add an external fan (that does not have PWM inputs) to help keep your Raspberry Pi cool. It's a basic transistor-driven motor controller, not a full bidirectional driver. It can be controlled via the Raspberry Pi's GPIO fan control option, found in the latest Raspberry Pi OS.



LumosRing

BRADÁN LANE STUDIO ♦ \$65 | hsmag.cc/lr

The more LEDs the better! And the LumosRing, designed by Bradán Lane STUDIO, certainly lives up to that expectation. Featuring over 250 RGB LEDs, this beautifully designed black PCB makes for a great clock, visualizer, or you can even use it to play games. And fortunately, you don't have to solder all those LEDs, as the LumosRing comes fully assembled.

The 240 LEDs on the outer rings lend themselves to making a perfect analogue clock. Coupled with the 70 LEDs configured in two 5x7 blocks in the centre, you can add extra functionality like a temperature display, or even a scrolling message or stylised icons. The board also features an included ESP23-S2 running CircuitPython. You can add arcade buttons for extra functionality, and a diffused faceplate to smooth out all those LEDs. Check out the product page (hsmag.cc/LumosRing) for a video demo of all those beautiful addressable RGB LEDs! 

VERDICT

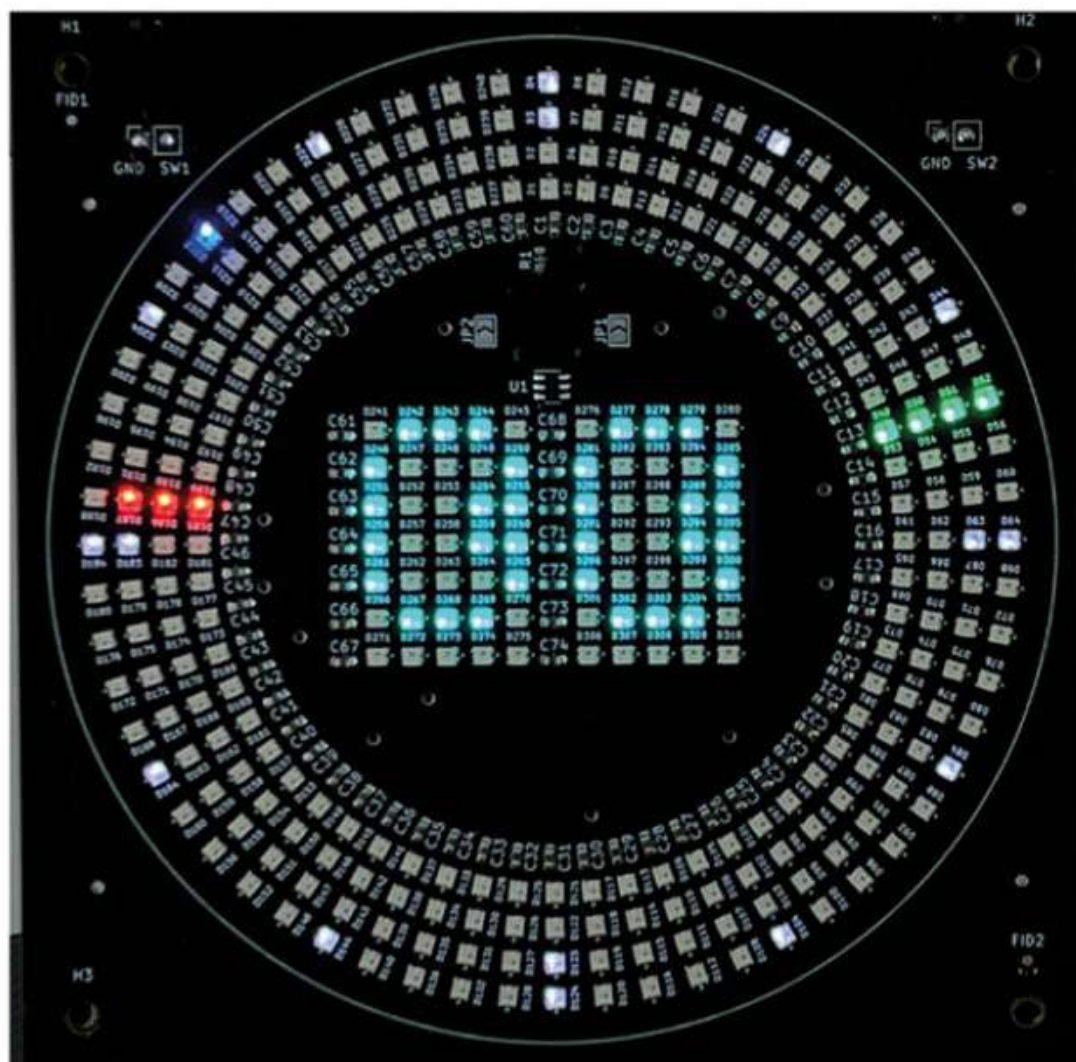
LumosRing

Hundreds of LEDs, on a beautiful PCB.

9/10



Below 
Squares or circles?
Why not both?



English not your mother tongue?

The MagPi is also available in German!



Subscribe to the German edition of The MagPi and get a Raspberry Pi Pico with headers and a cool welcome box **FOR FREE!**

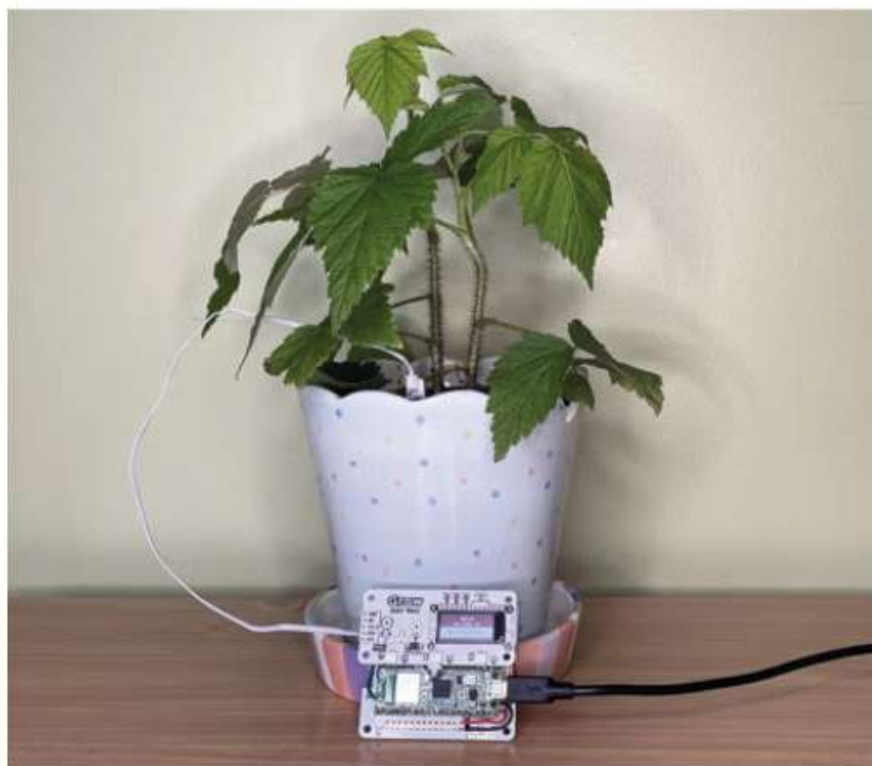
Use the coupon code **115PicoDE** on www.magpi.de/115



10 Amazing: Raspberry Pi Pico projects

Make a cool and small project with a Pico

Last month we revealed Raspberry Pi Pico W – Pico but with wireless LAN! There are loads of new ways to use Pico, and some projects you can make better with a Pico W. Here are just ten incredible examples of Pico projects to help inspire you. 📺



▲ Plants with personality

Mood monstera

This plant project makes use of a Pico W's wireless features to text you when it needs water, or generally when it feels like it.

magpi.cc/personalityplant



▲ Cyber glasses

Futuristic wearable

These 3D-printed glasses are more of a base for your own glasses ideas. However, they look great as they are with an LED ring focuser over one eye.

magpi.cc/cyberglasses

▶ Commodore 64 emulator

Retro computing

A C64 expansion board made from a Pico which actually helps the old computer work a bit better than it used to.

magpi.cc/c64pico

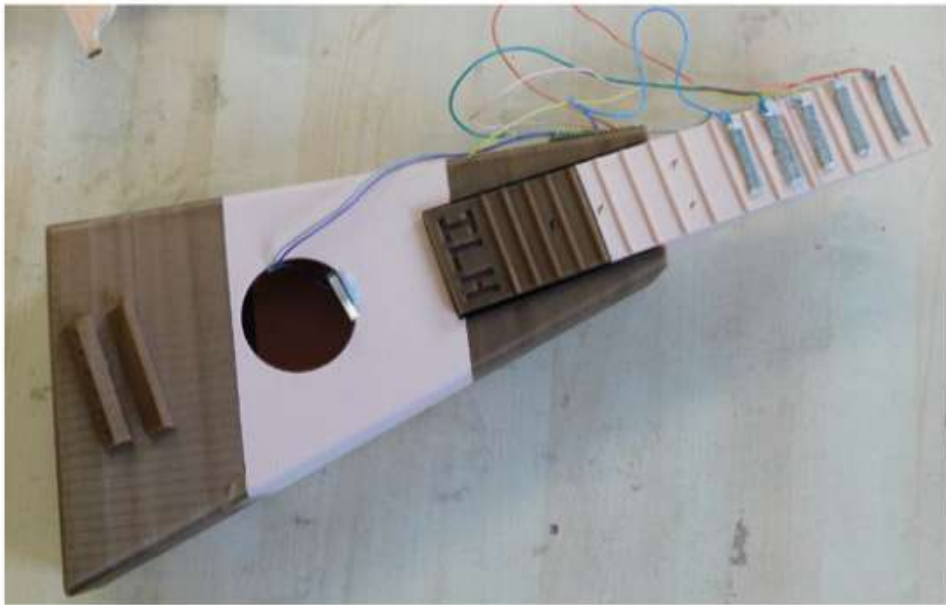


▲ Solar System Display

Space clock

You can use this great project to both tell the time, and also see the orientation of the planets at specific dates and times!

magpi.cc/picosolar



▲ Guitar games controller

Heroic guitar prowess

From the ashes of a broken 3D-printed guitar comes a controller from our friends at HackSpace Magazine to work with a game from our other friends at Wireframe.

magpi.cc/guitarcontroller



▲ Trill Guitar

Musically inclined

Using some special touch sensors and a custom-made body, you too can create this MIDI guitar.

magpi.cc/trillguitar

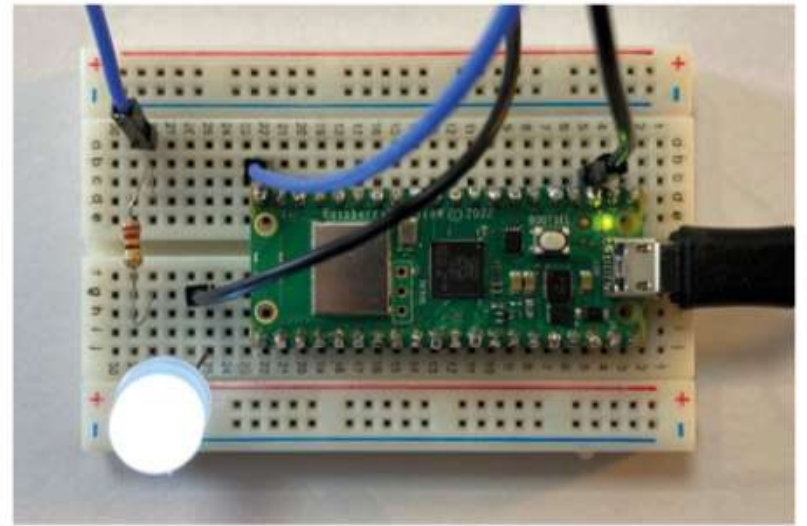


◀ Spooky activator

Light it up

This platform allows you to light up whatever is placed on it, and is very simple to make!

magpi.cc/spookactivate



▲ Run a web server on Pico W

Network control

A great example of how to use Pico W from Alasdair Allan of Raspberry Pi. It's incredibly practical too for remote control projects.

magpi.cc/picoserver



◀ Stream deck

What's up, gamers

This specific project uses a Keybow 2040, built on the same chip as Pico. However, you can use the code in the same way with some custom keys added to Pico.

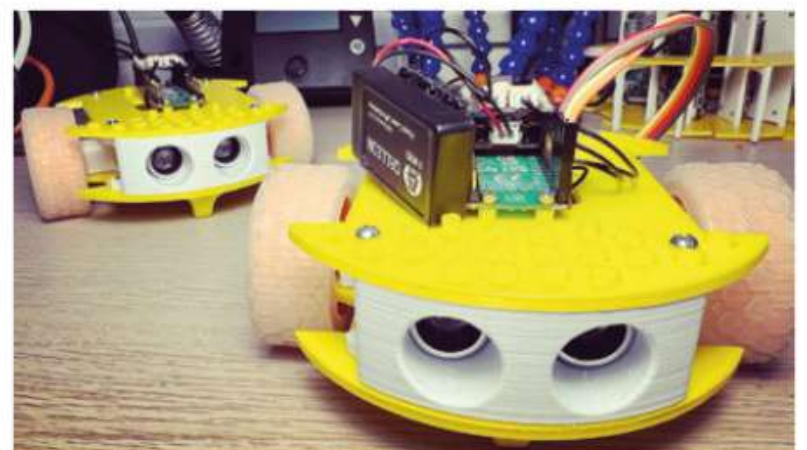
magpi.cc/111

▼ Upgraded Burgerbot

Wireless automated sandwich

Kevin McAleer's excellent Burgerbot is now able to be controlled wirelessly using a Pico W! The whole Pico-based robot build itself is also very impressive.

magpi.cc/burgerbotw



Learn microcontrollers with Raspberry Pi

Improve your knowledge of microcontrollers with these resources. By **Phil King**

Adafruit Learning Centre

AUTHOR

Adafruit

Price:
Free

magpi.cc/adafruitmcu


The smaller siblings of fully-fledged microcomputers, **microcontrollers are simpler programmable devices used to control electronics hardware.** They're often embedded into home appliances such as washing machines and microwave ovens, acting as their 'brains'.

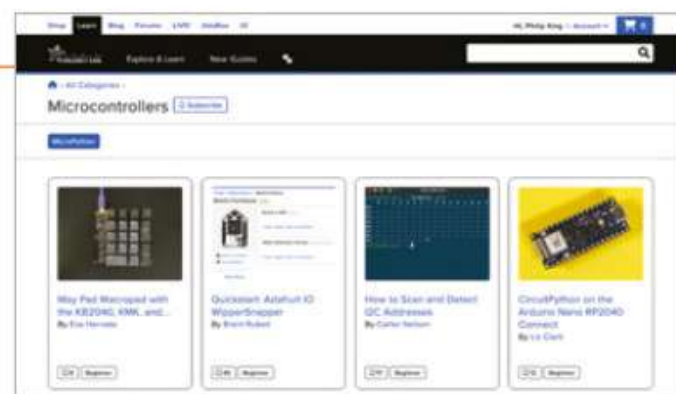
Strictly speaking, the microcontroller is the silicon chip, but it often sits on a PCB to break out its connections. In

the maker world, there are countless programmable microcontroller boards available; popular types include Raspberry Pi Pico and Pico W, Arduino, Teensy, and BeagleBone.

The Adafruit Learning Centre is a good place to learn about microcontrollers, as well as IoT and electronics in general. The microcontrollers section features a host of project tutorials based

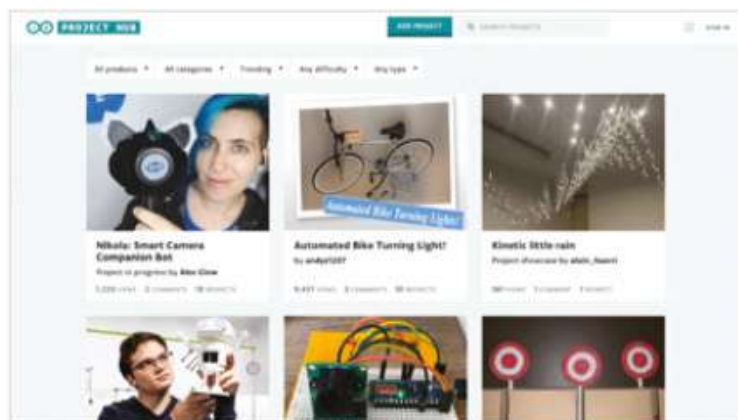
on various boards, including Adafruit's own products such as its Circuit Playground and Feather ranges.

There's also a handy 'How to Choose a Microcontroller' article that gives an overview of the boards available and how to select the right one for your project. 



Useful websites

Visit these sites for microcontroller projects and news



ARDUINO PROJECT HUB

Got an Arduino? Then check out the Arduino Project Hub to find countless projects – filterable by board, difficulty, etc. – complete with step-by-step instructions.

► magpi.cc/arduinohub

RASPBERRY PI PICO PROJECTS

The Raspberry Pi Foundation has a range of Pico project tutorials,

including a sensory gadget and sound machine, plus getting started guides for Pico and Pico W.

► magpi.cc/picoprojects

HACKADAY MICROCONTROLLERS

Get all the latest news about microcontrollers, along with some interesting projects to inspire you, on this Hackaday feed.

► magpi.cc/hackadaymcu

Get Started with MicroPython on Raspberry Pi Pico

AUTHOR


Gareth Halfacree & Ben Everard

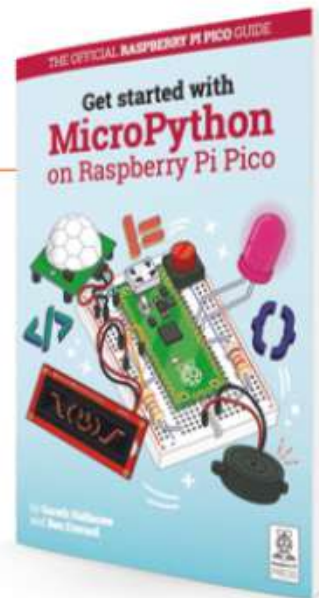
Price:
£10 (Free PDF)

magpi.cc/picobook

Pico is based on Raspberry Pi's own RP2040 chip, a powerful low-cost microcontroller also used in some other boards. While Pico can also be programmed with C/C++, this official guide book is all about coding it using MicroPython, a version of Python for microcontrollers.

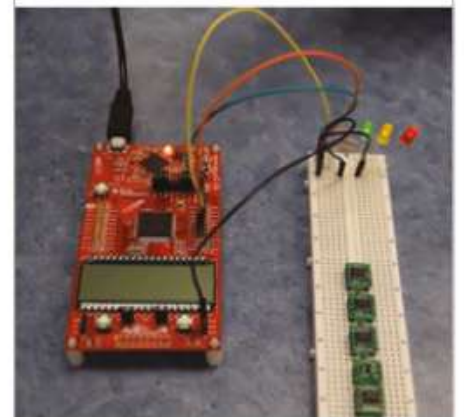
The first chapter shows how to set up your Pico, soldering pin headers, and installing MicroPython. After covering the basics of MicroPython and physical computing, the book

moves on to a series of projects. Explained in step-by-step detail, complete with wiring diagrams, these include a traffic light controller, reaction game, burglar alarm, temperature gauge, and data logger. The final chapter covers connecting an LCD using both the I2C and SPI protocols. An appendix explains Pico's unique Programmable IO with the example of controlling NeoPixel strips. 



Online courses

Learn microcontroller magic with these web courses



MICROCONTROLLERS AND THE C PROGRAMMING LANGUAGE

Create C programs for a TI MSP430 microcontroller using inputs/outputs, timers, ADCs, comm ports, and an LCD in this free Udemy course.

► magpi.cc/udemymcuc

INTERFACING WITH THE ARDUINO

This Coursera course teaches you how and when to use different types of sensors and how to connect them to an Arduino and program it.

► magpi.cc/arduinocoursera

RASPBERRY PI PICO AN INTRODUCTION WITH MICROPYTHON

After setting up Pico and learning some MicroPython coding basics, this Udemy course moves on to showing you how to use it to control a robot.

► magpi.cc/udemypico

Programming Arduino: Get Started with Sketches

AUTHOR


Simon Monk

Price:
£12 / \$15

magpi.cc/programarduino

Arduino microcontroller boards come in various shapes and sizes and there are also many clones. Whichever model you have, this book is one of the best to get started using it.

For the uninitiated, sketches are programs written in a modified C language in the Arduino IDE; the book teaches you how to write them using functions, arrays, and strings. It then moves on to cover interaction with electronics via the Arduino's digital and analogue I/O pins for electronics projects. Subsequent chapters deal with the standard Arduino library functions, data storage, interfacing with displays, and creating a web server.

There's also an optional Udemy course based on the book. Once you've mastered the basics, a follow-on book, *Programming Arduino Next Steps*, covers more advanced sketches and projects. 





Matt Richardson

The previous head of the Raspberry Pi Foundation North America has now joined the commercial arm to keep in touch with the community

- Name **Matt Richardson** | ➤ Occupation **Community engagement manager**
- Community role **Raspberry Pi liaison** | ➤ Twitter **@MattRichardson**

If you've ever been to an event in the US where Raspberry Pi was officially there, whether hosting the event or just with a stall, chances are you've met **Matt Richardson**. His work spreading Raspberry Pi around North America has been huge. Now though, he has a new job, albeit still with Raspberry Pi. He's a maker at heart, however.

"Before Raspberry Pi came about, I stumbled on the maker community because Arduino projects were starting to surface

in the technology blogs that I read at the time," Matt tells us. "People were making cool stuff, and it inspired a ton of ideas for things I wanted to make. I ordered an Arduino starter kit, and I just took to it immediately. I made a lot of things like weird doorbells, wireless gum ball machines, and internet-connected scrolling LED message signs. I eventually started posting my own articles and how-to videos. Eventually I got noticed and became a contributor for Make Magazine."

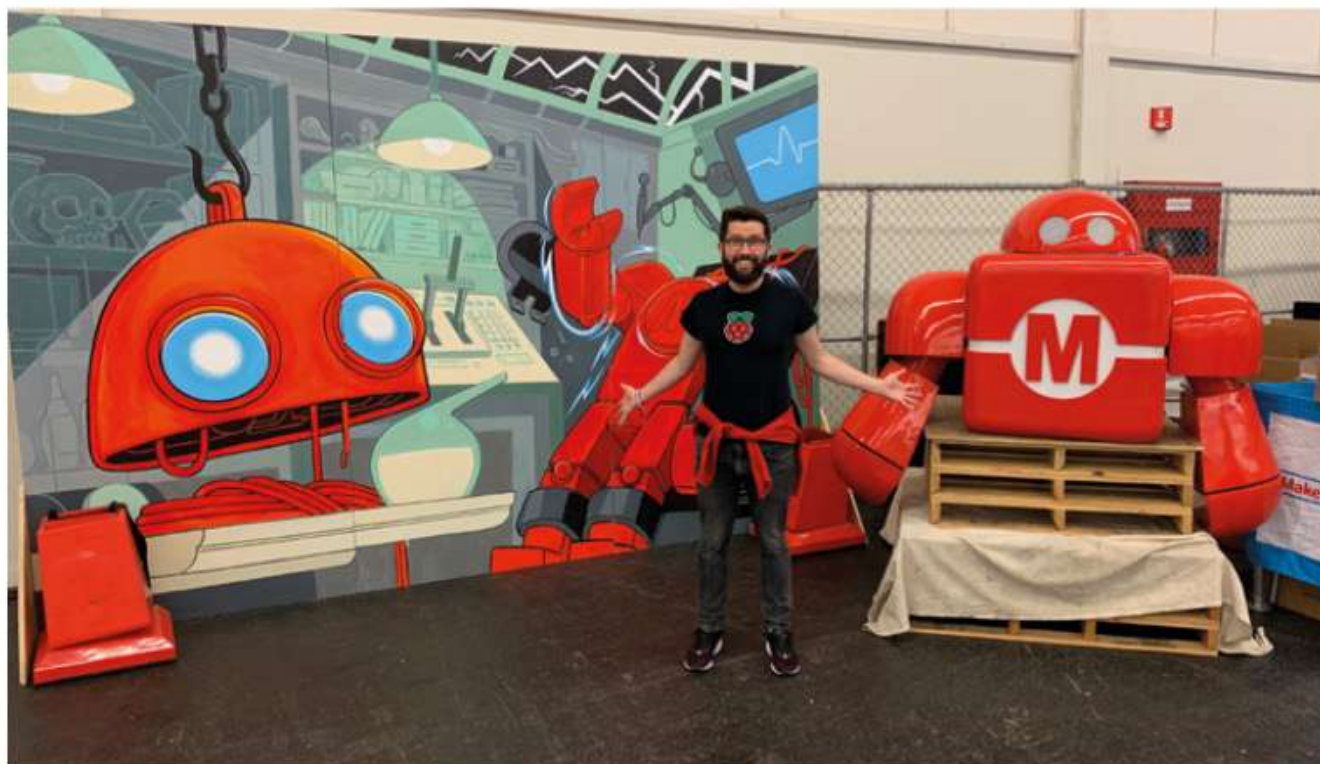
When did you learn about Raspberry Pi?

I remember being on a conference call for Make editors and one of my colleagues mentioned that this new thing called Raspberry Pi was making headlines – this was back before it started shipping. I was sceptical that we'd ever see it ship at the promised \$35. Boy was I wrong, and I couldn't be happier that I was wrong about that! I emailed Eben Upton to see if he could send me one to review for Make and amazingly, he did! I booted up this Raspberry Pi connected to my family room TV and I knew that this was going to be HUGE. A few years later, I started working for Raspberry Pi, for some of that time on the commercial side, and for some of that time on the non-profit side. I'm lucky that I've had the chance to gain experience in so many facets of the organisation.

What is the American community like?

Although the internet was an important catalyst for the maker movement and the Raspberry Pi community, I think both have really benefited from in-person

▼ Matt represented the Raspberry Pi Foundation at many Maker Faires





◀ Helping people take their first steps into coding at New York World Maker Faire

▼ The dynamic headlight in use by Matt

“ I'm lucky that I've had the chance to gain experience in so many facets of the organisation ”


engagements like Maker Faires and Raspberry Jams. Whether it was a small gathering of five people at a makerspace, or a multi-day event for thousands of people, these events really helped to grow and strengthen the community. The pandemic had a detrimental effect on that, but with things getting back to a more normal state, I think we're starting to see opportunities for new events to sprout.

What kind of events do you like to attend?

I love geeky, family-friendly events like Maker Faires and Raspberry Jams. You get to meet such an incredible spectrum of people, some who've never heard of Raspberry Pi and have never written a line of code, to engineers who've been tinkering with Raspberry Pi in their spare time since day one. I especially

love seeing a parent sitting at a Raspberry Pi with a child in their lap, showing them how to code and light up an LED. It reminds me of how I first learned to code with my father in the 1980s.

What is your favourite thing you've made with a Raspberry Pi?


The one I had the most fun making was the dynamic bicycle headlight. I strapped a battery-operated projector to the handlebars of my bicycle and wired up a Raspberry Pi to read a sensor on the wheel and display my speed in the projector's beam down ahead of me. The best part was when I was troubleshooting it, I had to log into its Raspberry Pi wirelessly via SSH. I thought it was crazy to be remotely logging into my own bicycle to get diagnostics and upload new code. 



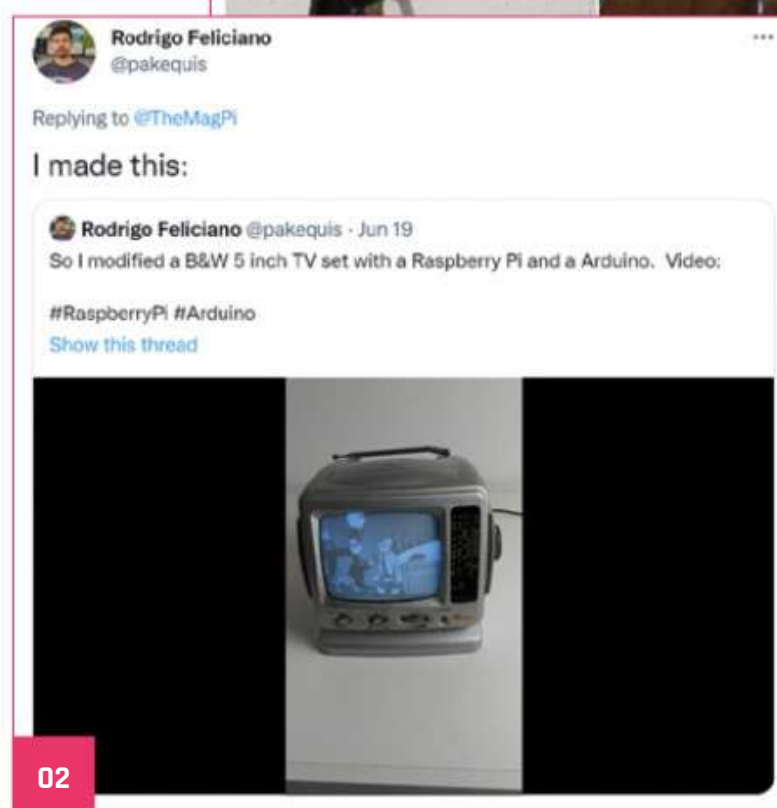
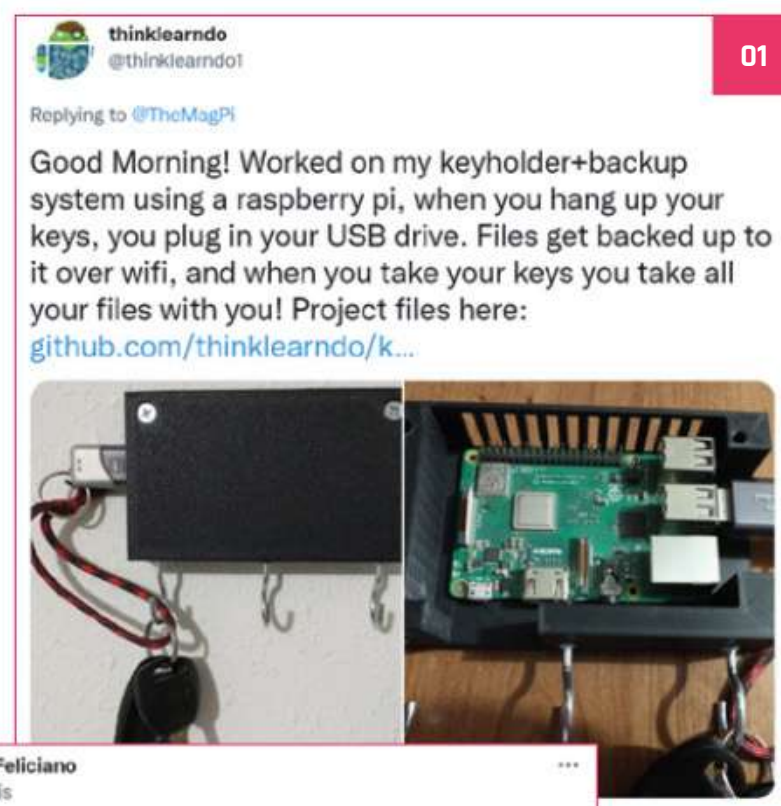
MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday! 

01. This is a very cool and useful idea!
02. Be careful when working with CRT TVs – this is great though
03. PiMowBot Model D looking very sharp
04. A very cute robin for you in the pages of *The MagPi*
05. We believe this is out now, if you want to check it out!
06. The AR future is here and we want in
07. This is a cool little conversion script
08. This could be a very dangerous game to play, depending on the ABV
09. There is a lot packed into this little project
10. This is pleasingly retro, we'd love to see the deck it's connected to
11. This allows for target ranging and looks very neat





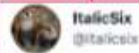
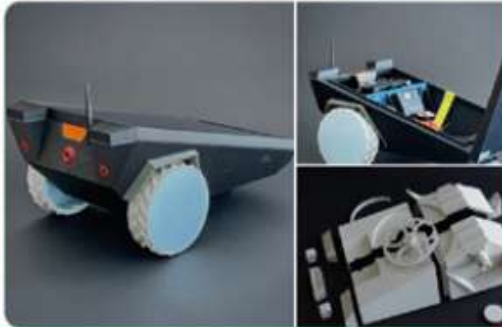
03

Replying to @TheMagPi

TimE just finished the revision of PiMowBot 3D files. The new, smart looking Model D is now available.

Visit bit.ly/2YxRoYw @Cults3D to download the new 3D-models.

#make #3Dprinting #IoT #smart #robotics #lawnmower with #RaspberryPi inside.



04

Replying to @TheMagPi

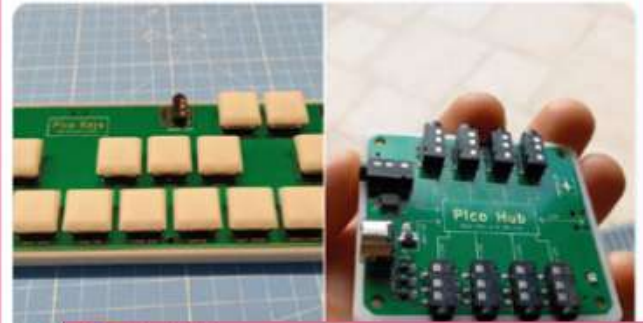
Bit of bird feeder action on YouTube
youtu.be/laa_B6kctOE



05

Replying to @TheMagPi

Making progress on the Pico Keys MIDI keyboard/synthesizer and the Pico Hub progammable MIDI hub 🎹 🎵 🎛️



Replying to @TheMagPi

I'm building a random beer selector for my friends bar.... It'll be hung above the taps once I finish the woodwork frame, and the button is mounted into a mini keg that will sit on the bar.



Replying to @TheMagPi

I made a Pi powered wearable HUD :)



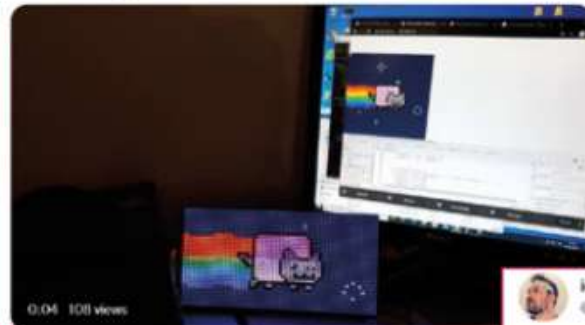
06



07

Replying to @TheMagPi

Did I ever post the picture of the browser->LED bridge I'm building?

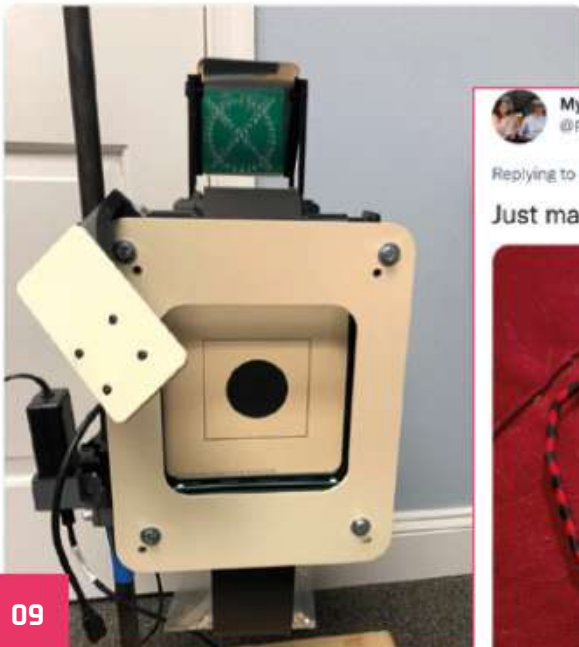


08



Replying to @TheMagPi and @Raspberry_Pi

This is Athena, our Pi powered EST system now at over 200 ranges.



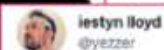
09



10

Replying to @TheMagPi

Just made this Aviator "patch" cable for a cyberdeck

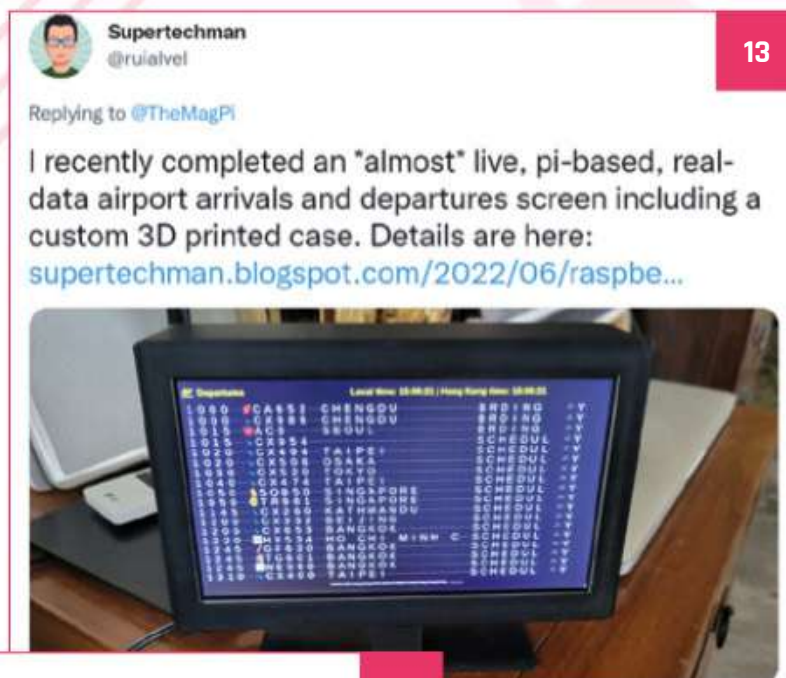


Replying to @TheMagPi

An AI terminal using Raspberry Pi, ArduCam, ReSpeaker, Coral TPU, PicoVoice, and more..



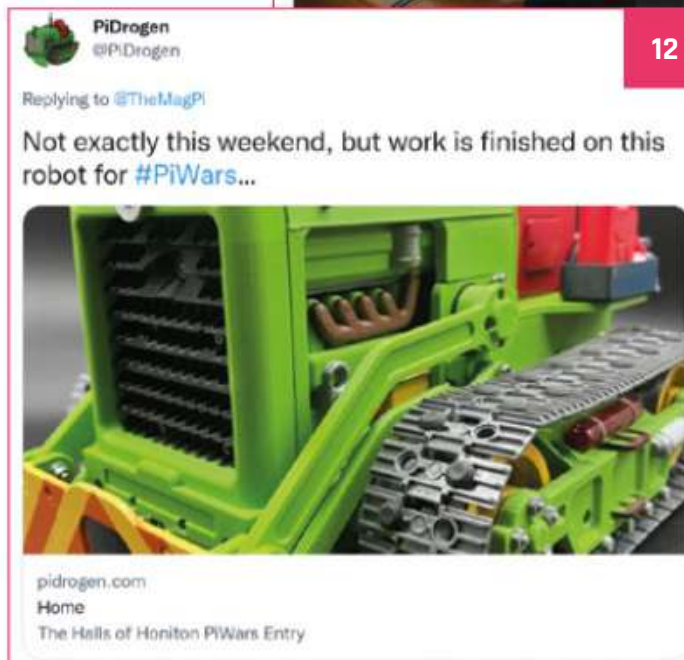
11



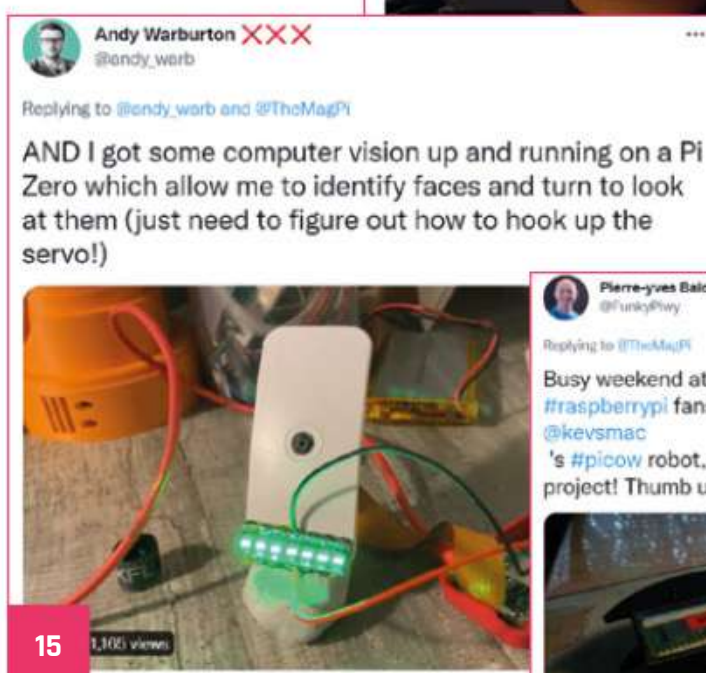
13



14



12



15



17

12. Tractor robot is a very cool idea
13. If you're making a lot of flights, then this is essential
14. This robot is very cute and gives us a little retro vibe
15. The extra LEDs on this make it a little more interactive, it's good
16. We once saw a Big Mouth Billy Bass voice assistant and this is just as cool
17. Great to see the Burgerbot get built elsewhere
18. This is a really cool upcycling project!
19. The new Jurassic Park film has interesting special effects we see



19

Crowdfund this!

Raspberry Pi projects you can crowdfund this month



16



Lyra+

We reviewed the original Lyra many issues ago, and thought it was pretty good – now with Compute Module 4 being a thing, the makers at Creocode are crowdfunding a new version called Lyra+. It's compatible with many game streaming services such as Steam Link, xCloud, PlayStation Remote Play, and others.

► magpi.cc/LyraPlus



18



Revocode

This modular arcade machine allows you to change out decks, from classic joystick and buttons to steering wheels and pedals, and even paddle and trackball controls. It also has cool changeable marquees at the top to suit your home arcade's aesthetic.

► magpi.cc/Revocode

Your Letters



Unseasonably early

As August is nearly here, I've been starting to sort out my Halloween plans. Would it be too early to send my WIPs and other bits and pieces to you on MagPi Monday?

Arin via email

Absolutely not – we love to see how projects get built, and it's a good idea to get them started early. On the internet especially, Halloween starts in early August for many, so it would not be out of place.

We'll also be doing our annual Halloween feature and post-October roundup, so anyone else making a Halloween project should also send us photos via email, Facebook, or on Twitter. We'd love to see!



▲ The run-up to Halloween gets longer and longer, but no less spooky

Pico Stock

I was excited to hear about Raspberry Pi Pico W when I read about it last issue. However, I was wondering if it might have some stock problems like other Raspberry Pis? Does the chip shortage affect it as well?

Angela via email

Pico W is made using RP2040, which is a Raspberry Pi-made chip, which allows the commercial arm of Raspberry Pi to more easily manufacture and sell Pico, Pico W, and RP2040s in general. At launch, some places sold out over a few days, but stock is available again.

You can also get one with a global print subscription, offers starting at £10 for the UK! Check them out at magpi.cc/subscribe.



▲ Raspberry Pi Pico W will not experience the same issues as other Raspberry Pi models

Contact us!

- Twitter [@TheMagPi](https://twitter.com/TheMagPi)
- Facebook magpi.cc/facebook
- Email magpi@raspberrypi.com
- Online forums.raspberrypi.com



▲ We hope to bring you more pieces about events as they start up again

More events

I enjoyed reading up about EMF Camp in the last issue. What a wonderful show! It made me miss attending Raspberry Jams, and I realised that you don't list the events anymore. Will that be coming back any time soon?

Chu via Twitter

Due to world events, we thought it was best not to advertise in-person events about 18 or so months ago – they were mostly online at that point as well. We hope to bring it back soon, especially as Raspberry Pi representatives are attending events around the world at the moment. At the very least, look out for some reports from some events in future issues.

3 ISSUES FOR £10



FREE
RASPBERRY PI
PICO W

☎ Subscribe by phone:
01293 312193

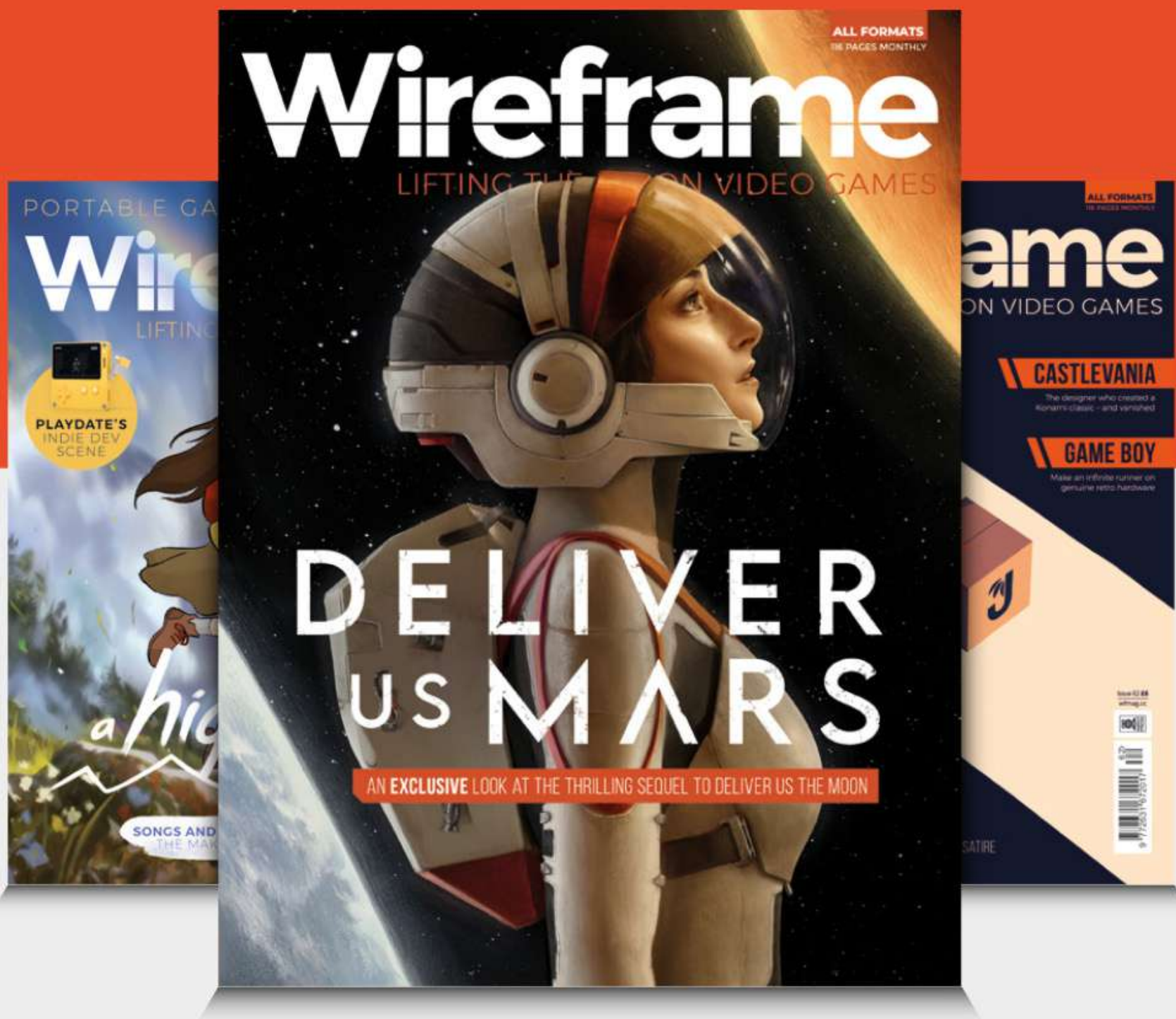
📧 Subscribe online:
magpi.cc/subscribe

✉ Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £30 every six months unless cancelled. Free Pico W included with a 12-month subscription in USA, Europe and Rest of World. This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

Wireframe

Join us as we lift the lid
on video games

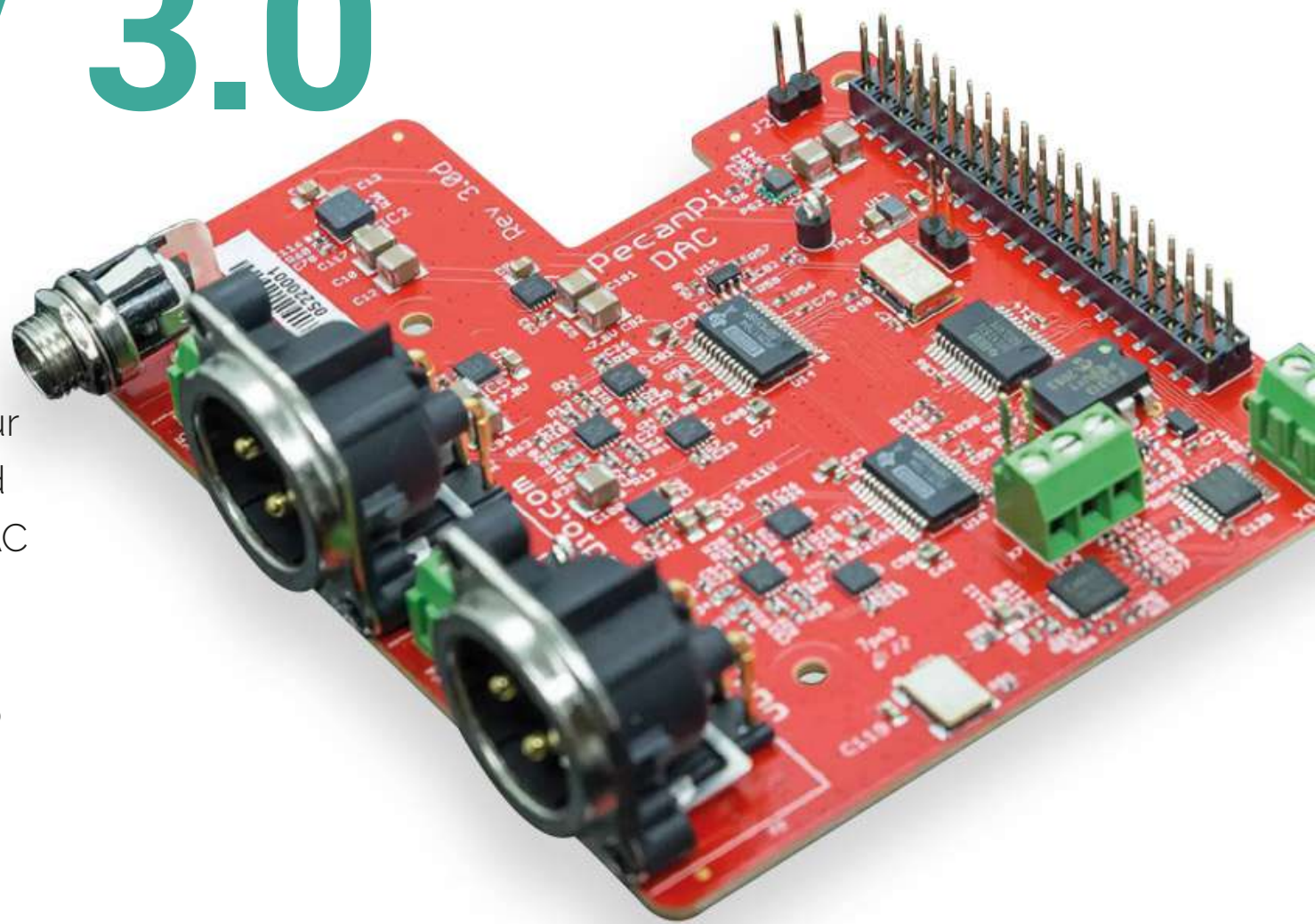


Visit wfmag.cc to learn more

WIN

PECANPI DAC REV 3.0

We reviewed the full PecanPi streamer a few issues ago, and our audiophile writer loved it. It's powered by a DAC expansion board that works with Raspberry Pi, and we have one to give away.



Head here to enter: magpi.cc/win | Learn more: magpi.cc/pecanpidac

Terms & Conditions

Competition opens on **27 July 2022** and closes on **25 August 2022**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter or any other companies used to promote the service.

HackSpace

TECHNOLOGY IN YOUR HANDS

THE **MAGAZINE**
FOR THE **MODERN MAKER**



SUBSCRIBE AND
SAVE UP TO
35%
on the cover price



ISSUE #57

OUT NOW

hsmag.cc

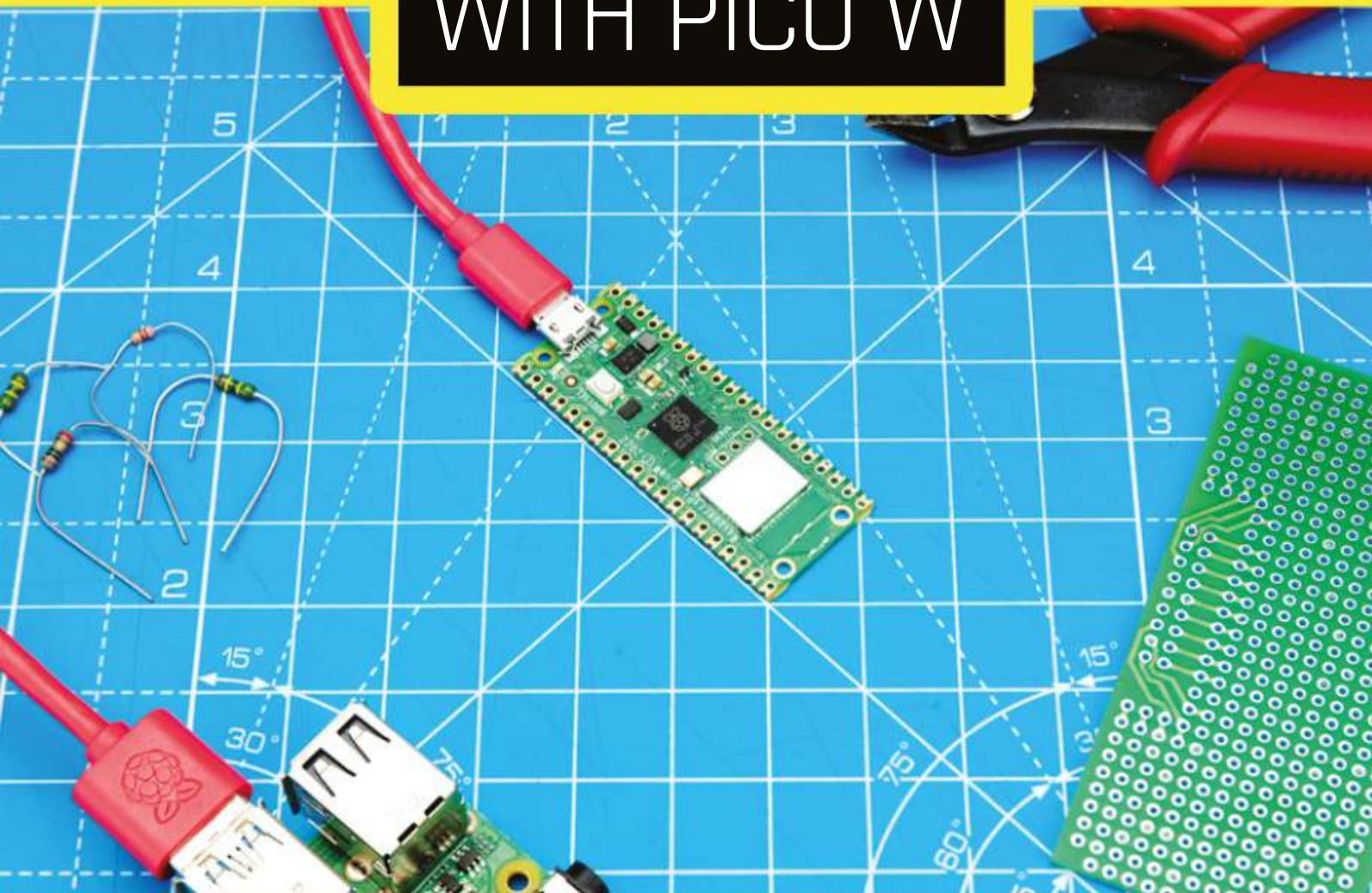


Available on the
App Store



GET IT ON
Google Play

LEARN ELECTRONICS WITH PICO W



THE MAGPI #121
ON SALE 25 AUG

Plus!

Prep for
Halloween

Train a Peto
Bittle dog

Run a Pico
web server

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER @TheMagPi

FACEBOOK fb.com/MagPiMagazine

EMAIL magpi@raspberrypi.com

EDITORIAL

Editor

Lucy Hattersley
lucy@raspberrypi.com

Features Editor

Rob Zwetsloot
rob@raspberrypi.com

Sub Editor

Nicola King

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Olivia Mitchell, Ty Logan,
Sam Ribbits

Illustrator

Sam Alder

CONTRIBUTORS

David Crookes, PJ Evans,
Rosemary Hattersley, Nicola
King, Phil King, Stephen Smith,
Marc de Vinck

PUBLISHING

Publishing Director

Russell Barnes
russell@raspberrypi.com

Director of Communications

Liz Upton

CEO

Eben Upton

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk



This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under



a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2051-9982.



Making makers

And makers making a difference with
Raspberry Pi Pico W. By **Lucy Hattersley**

Raspberry Pi makes a lot of things: computers, Pico microcontroller boards, RP2040 microcontroller chips, cameras, touchscreen displays, and, of course, magazines.

What we're really trying to make though, is a difference.

The world is full of people who have lost touch with the mechanical roots of technology. They use shiny devices made of glue and glass, with components sealed away and moving parts that are only visible at the microscopic level. They lack heart and soul, and any sense of wonder or joy.

The word 'algorithm' may as well be 'abracadabra' for most people. Tech is indistinguishable from magic.

Raspberry Pi Pico has no glass and glue hiding the circuitry. The RP2040 processor is on full display. It's almost asking you to make something with it.

What you make with a new Pico W is, of course, up to you. There's a world of possibilities, from home automation projects to personal robotics; its ultra-small footprint and low power consumption make it ideal for wearable projects and embedding on your person or upcycling old equipment with ultra-modern features.

What you are really creating though is a little more personal understanding of the world around you. When delivery robots roll down the street, you'll see

the motors and GPS unit inside; when supermarkets get rid of checkouts, you'll see the image detection cameras and the micro-location technology. When social media algorithms start to push you towards a more extreme edge of your favourite hobby, you'll – well, we hope – know enough to step back.

The difference between knowing and not knowing how technology works can

“ We get to start again with all those projects we love ”

be life-changing stuff. *The MagPi* has been using Raspberry Pi for years to make this difference, in our own small way, to the people who pick up our magazine.

Pico power-up

For the rest of this year we're going to make the most of Pico W. We're going to start again, from the ground up.


From basic electronics to smart home kits, retro gaming and robotics. We're going to really go for it all over again. This time with the new Pico W.

Raspberry Pi Pico is a relatively new component in the Raspberry Pi products catalogue. But with Pico W's addition of wireless LAN networking and in-house silicon, it's clearly a

product having its moment in the spotlight. For various reasons, Pico W remains largely free from the broader supply chain problems. You can get Pico W in plentiful supply and should be able to throughout the rest of the year. So, now is the perfect time to pick up a Pico W and start making with it.

I've already got an electronics kit from Pimoroni to play around with, and a smart home automation system. Every one of *The MagPi*'s editorial team and freelancers have been asked to line up Pico W projects and tutorials. I can't wait for you to read about what they are working on.

Pico W is, of course, a great thing to happen for Raspberry Pi. But it's also a great thing for *The MagPi*. We get to start again with all those projects we love.

It's also time for us to learn to make a difference all over again. As well as being an incredible engineering tool, Raspberry Pi teaches children – of all ages – how computers work, and how to use computers to make a difference in their lives. 

Lucy Hattersley

Lucy is editor of *The MagPi* magazine and is currently melting in 30°C temperatures. She is on the verge of inventing Pico-flavoured ice cream.

magpi.cc

HIGHPI PRO

———— The new case from the HiPi.io team ————



- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!



PiKVM HAT
for DIY and custom projects



Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers



Reseller suggestions and inquiries:
wholesale@hipi.io